

# Multi-fluid solver for plasma simulation in a Hall-Thruster engine



**POLITECNICO**  
**MILANO 1863**

Msc in Computational Science and Engineering

(NUC) Mathematical and Physical Modeling for Nuclear Application

Relatore: Paolo Francesco Barbante

Correlatore: Thierry Magin

Tesi di Laurea di:  
Lorenzo Vallisa, matricola 904657

Anno Accademico 2019-2020





# Contents

List of Figures	x
List of Tables	xiii
List of Symbols	i
<b>1 Introduction</b>	<b>1</b>
1.1 Principles of Hall-Thrusters . . . . .	1
1.2 Final goal of the thesis . . . . .	4
<b>2 Governing equations for multi-fluid model</b>	<b>5</b>
2.1 Kinetic description . . . . .	5
2.2 Derivation of Euler set of equations . . . . .	6
2.3 Source terms derivation . . . . .	12
2.3.1 Collisional source terms . . . . .	12
2.3.2 Electromagnetic field . . . . .	18
2.3.3 Full set of dimensional equations . . . . .	19

2.4	Nondimensionalization . . . . .	20
<b>3</b>	<b>Numerical tools</b>	<b>22</b>
3.1	Finite Volume Method for 1D problem . . . . .	22
3.2	Boundary conditions . . . . .	23
3.3	Numerical fluxes . . . . .	24
3.4	Temporal discretization . . . . .	25
3.4.1	Midpoint Euler . . . . .	25
3.4.2	Backward Euler . . . . .	26
3.5	Poisson's solver equation . . . . .	29
3.6	Acceleration techniques . . . . .	31
3.6.1	Dual-Time-Stepping . . . . .	32
<b>4</b>	<b>Implementation and verification</b>	<b>39</b>
4.1	<i>FORMICA</i> : Multi-fluid 1D Solver for collisional charged plasma	39
4.2	Verification of multi-fluid model: 1D Sod-Shock . . . . .	41
4.3	Verification of Poisson's solver: dispersion relations for longitudinal modes in plasma . . . . .	43
<b>5</b>	<b>Results</b>	<b>47</b>
5.1	Numerical estimates of reaction rates . . . . .	47
5.2	Plasma expansion into vacuum . . . . .	50

5.3	Hall-Thruster engine without magnetic field: performance analysis on DTS method . . . . .	58
5.4	Preliminary results for Hall-Thruster engine with radial magnetic field $\mathbf{B}$ . . . . .	62
<b>6</b>	<b>Conclusions</b>	<b>69</b>
6.1	Future works . . . . .	70
	<b>Appendices</b>	<b>73</b>
.1	Source term derivation . . . . .	74
.2	LIS library . . . . .	77
.3	Predictor Corrector . . . . .	80
.4	LAPACK library . . . . .	81

# List of Figures

1.1	PPSR 1350-G . . . . .	2
1.2	Sketch of Hall-Thruster engine . . . . .	3
1.3	Trap effect of magnetic field . . . . .	4
2.1	Isotropic scattering in case of still neutrals . . . . .	13
3.1	1D Mesh discretization . . . . .	23
3.2	Midpoint Euler vs Forward Euler scheme . . . . .	26
3.3	Block tridiagonal matrix . . . . .	28
3.4	Chebyshev nodes . . . . .	30
3.5	Electrons-Ions in Dual-Time-Stepping . . . . .	32
3.6	Different choice of Newton's parameter for strongly unsteady and weakly unsteady solution patterns . . . . .	35
3.7	Local pseudo-timestep . . . . .	36
4.1	Physical iteration summary (DTS) for nondimensional (ND) simulation . . . . .	40
4.2	FORMICA electrons Sod-Shock . . . . .	42

4.3	FORMICA ions Sod-Shock . . . . .	42
4.4	Cold-Plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT . . . . .	44
4.5	Hot-Plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT . . . . .	45
4.6	<b>B</b> into plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT . . . . .	46
5.1	LHS: Energy range check for integrand RHS: $\sigma$ and Maxwellian plot . . . . .	48
5.2	Ions - neutrals elastic scattering . . . . .	48
5.3	Excitation . . . . .	49
5.4	Ionization . . . . .	49
5.5	Electrons - neutrals elastic scattering . . . . .	50
5.6	Electrons and ions density evolution . . . . .	53
5.7	Electric field and ions axial speed . . . . .	54
5.8	LHS Electrons temperature profile in space and time RHS Average temperature in time at ions expansion front . . . . .	54
5.9	Scaled electric field front: results for Midpoint Euler (also with $T_{el}$ imposed) and implicit scheme (DTS) . . . . .	55
5.10	Scaled ions velocity front: results for Midpoint Euler (also with $T_{el}$ imposed) and implicit scheme (DTS) . . . . .	56
5.11	Sensitivity analysis electric field front . . . . .	56
5.12	Sensitivity analysis ions velocity front . . . . .	57
5.13	Hall-Thruster with no <b>B</b> : boundary and initial conditions . . . . .	58



5.14	Ions axial velocity: ME vs DTS . . . . .	60
5.15	Electric Field: ME vs DTS . . . . .	60
5.16	Ions Pressure: ME vs DTS . . . . .	61
5.17	$\mathbf{B}$ profile . . . . .	62
5.18	Xenon gas steady-state profile . . . . .	63
5.19	$\rho_i$ in <i>Case 1</i> . . . . .	64
5.20	$P_i$ in <i>Case 1</i> . . . . .	65
5.21	$\mathbf{E}$ in <i>Case 2</i> . . . . .	65
5.22	$\phi$ and $\mathbf{E}$ in <i>Case 3</i> . . . . .	66
5.23	$\rho_i$ and $u_i$ in <i>Case 3</i> . . . . .	67
1	Data structure of BSR . . . . .	77

# List of Tables

4.1	FORMICA input settings . . . . .	41
4.2	Electrons BC Sod-Shock . . . . .	41
4.3	Ions BC Sod-Shock . . . . .	41
5.1	Plasma expansion into vacuum: BC . . . . .	52
5.2	Plasma expansion into vacuum: numerical settings . . . . .	52
5.3	DTS parameters setting for no <b><i>B</i></b> Hall-Thruster testcase . . .	59
5.4	Different numerical settings for DTS sub-iterations . . . . .	59
5.5	Initial and boundary conditions for <b><i>B</i></b> Hall-Thruster <i>Case 1</i> (electrons:ions) . . . . .	63
5.6	Restart settings for Hall-Thruster simulation with magnetic field	68
5.7	Time performances for restarted DTS in Hall-Thruster test case without magnetic field . . . . .	68

## List of Symbols

$\delta_{ij}$	Kronecker's delta
$\Omega_c$	Cyclotron's frequency
$\omega_{pe}$	Electrons plasma frequency
$\omega_{pi}$	Ions plasma frequency
$k_B$	Boltzmann constant
$\tau$	Pseudo timestep
$\mathbf{E}$	Electric field
$\mathbf{B}$	Magnetic field
$\lambda_D$	Debye length
$q_j$	Particle charge of j-th species (Coulomb in SI)
$m_j$	Mass of j-th species particle
$\epsilon$	Dielectric tensor
$\epsilon_{ex}$	Activation energy for excitation
$\epsilon_{iz}$	First ionization energy
$\nu_{iz}$	Ionization frequency
$\nu_{el}$	Electrons-Neutrals elastic collision frequency
$\nu_{il}$	Ions-Neutrals elastic collision frequency
$\nu_{ex}$	Excitation frequency
$e$	Internal energy
$E$	Total energy
$P$	Fluid pressure
$J_i$	Current density of i-th population
$\phi$	Electrostatic potential
$N_D$	Number of particles in Debye sphere
$CLF_N$	Numerical CFL for electrons computed within the pseudo-timestep
$e_N$	Neper number: 2,718...





# Chapter 1

## Introduction

This thesis project was developed during an internship (in remote) at the von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium.

In this project a multi-fluid modeling approach for plasma behavior within a Hall-Thruster engine is presented. This non-conventional approach in literature may reveal some further insights when studying dynamics of electrically charged populations and the way they interact with neutral gas atoms in background. A first short introduction on Hall-Thruster engines is followed by a thorough mathematical derivation of the set of fluid equations solved for each charged species: electrons and ions. Analytical and numerical derivation of source term is then treated, followed by a presentation of the main numerical scheme adopted in the software and the algorithms proposed to improve its computational performance.

### 1.1 Principles of Hall-Thrusters

Hall-Thrusters were originally envisioned in the U.S. and in Russia about 50 years ago, with the first working device reported in America in the early 1960s. Recent mission by the European Space Agency (*ESA*) has spacecrafts mainly propelled through this kind of technology: **PPSR 1350-G** produced

by Safran Aircraft Engines and chosen by ESA to power SMART-1 (Fig. 1.1). Hall-Thrusters consists of a cylindrical channel with an interior anode,

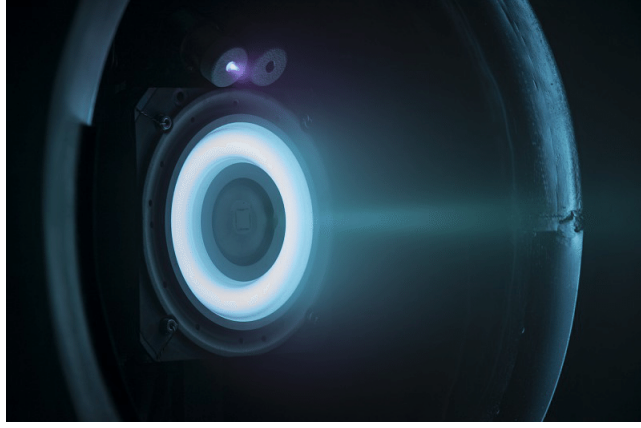


Figure 1.1: PPSR 1350-G

a magnetic circuit that generates a primarily radial magnetic field across the channel, and a cathode external to the channel, as quickly outlined in Fig. 1.2. Axial direction is the one alongside the engine channel (X axis in Fig. 1.2) whereas azimuthal direction follows the circular crown inbetween the two magnetic poles described in Fig.1.2. The apparent simplicity of this engineering devices hosts one of the most complicated, and for some aspects still well not-understood, physics within the field of plasma. An electric field is established between the anode, at the base of an annular channel, and the hollow-cathode, whose aim is to accelerate plasma produced outside of the thruster-channel (discharge region). The anode is a ring collocated at the base of the channel, right where gas is injected into the engine. Most of thrusters use Xenon gas, due to its high atom mass that strongly contributes to enhance the thrust performances. The main goal of radial magnetic field is to prevent electrons, coming from the current discharge at the cathode, from reaching the anode. Electrons are thus trapped along the magnetic field lines, around which they start to spin following the so called gyroscopic motion dictated by the cyclotron frequency:

$$\Omega_{c,e} = \frac{q_e \mathbf{B}}{m_e} \quad (1.1)$$

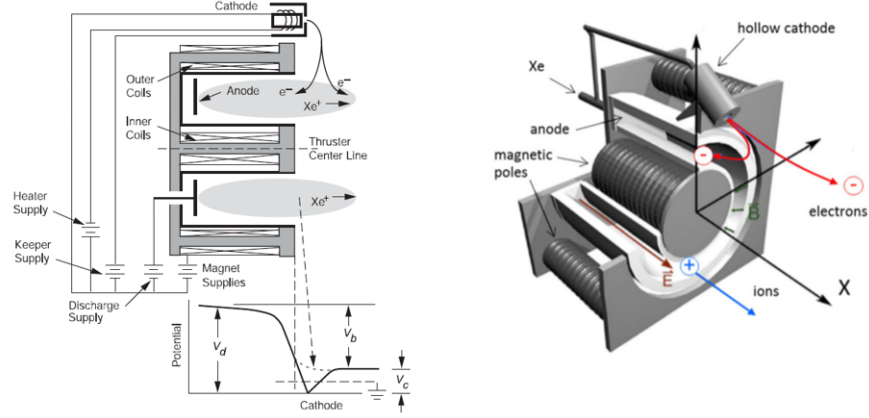


Figure 1.2: Sketch of Hall-Thruster engine

Electrons conductivity is further reduced by the interaction of the magnetic field with the electric field, they will in fact start to drift along the azimuthal direction due to the  $\mathbf{E} \times \mathbf{B}$  drift effect. Trapped electrons then efficiently ionize the injected propellant, giving birth to ions, whose much higher mass w.r.t. that of electrons allows them to not be trapped by magnetic field lines, but rather to be accelerated by the growing electric field: at the region of maximum magnetic field in fact, current continuity implies an increase of the electric field where the electrons conductivity is reduced. The Larmor radius, depicting the radius of the circles described by the helicoidal motion of charged particles around magnetic field lines, is in fact much higher for ions ( $\approx$  meters) than the one of electrons ( $\approx$  millimeters).

$$\rho_L = \frac{v_{\perp} m}{|q| B} \quad (1.2)$$

In literature [13] it is often referred to *Ionization Region* and *Acceleration Region* in order to distinguish between those two phenomena: the first one is usually located a bit closer to the anode than the latter one, since first ions are generated inside the thruster by ionization phenomena, and only then they feel the electric field that accelerates them outside the thruster. Ions and electrons eventually diffuse by collisional processes and electrostatic fluctuations to the anode and channel walls. A brief sum-up is sketched in Fig. 1.3. The details of the channel structure and magnetic field shape



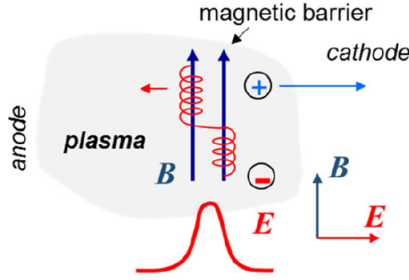


Figure 1.3: Trap effect of magnetic field

determine the performance, efficiency and life of the engine: the efficiency and specific-impulse of flight-model Hall-Thrusters are typically lower than those achievable in simple Ion-Thrusters, but the thrust-to-power ratio is higher and the device needs less power supply to operate. The life of Hall-Thrusters in terms of hours of operation is usually shorter than Ion-Thrusters, but the throughput is usually higher and the total impulse capability can be comparable [13].

## 1.2 Final goal of the thesis

Main goal of this thesis project is to create a software able to reproduce dynamics of ions and electrons in a 1D simulation by solving a set of Euler fluid equations. Interaction between the above mentioned charged species and neutral gas atoms will be studied through the computation of collisional integrals. The presence then of electric field and eventually magnetic field will be taken into account in order to reproduce the electromagnetic interaction between ions and electrons. Despite being a modeling approach new in the literature world, as far as Hall-Thrusters studies are concerned, the project aims at partially retrieve some of the classical results obtained for a steady-state simulation of an Hall-Thruster engine: indeed some key features of electric field and analogous profiles for ions density and pressure distributions. Another key aspect of this thesis will be to obtain results in an small computational time: in order to pursue this goal acceleration algorithms will be firstly introduced and then tested to improve performance of the software.

# Chapter 2

## Governing equations for multi-fluid model

### 2.1 Kinetic description

One of the most straightforward approach to model moving particles is indeed the Lagrangian one, in which the motion of every single particle is tracked using Newton's law. Despite being the most correct and precise approach, following each particle movement and interaction with external fields happens to be computationally unfeasible when the amount of particle becomes too large. One very first simplification is the one of switching from particle movement to particle phase space investigation: that is, when the number of objects to be investigated rises, a statistical approach is the answer. Instead thus to treat every single particle with its own Lagrangian variables, now particles are distributed according to a distribution function  $f_a$  for every  $a$  species, which is indeed a function of phase space, phase velocity and time. The formulation now is called kinetic, in which the unknowns are Eulerian fields and no more Lagrangian trajectories: more specifically now the unknown is the average<sup>1</sup> distribution function  $f_a(\vec{x}, t, \vec{v})$ . One of the the most famous equation within the field of kinetic description is the Boltzmann equa-

---

<sup>1</sup>usually distribution functions are averaged over a control volume in order to consider the average behavior of a certain amount of particles representative for that control volume

tion, in which the unknown is indeed  $f_a(\vec{x}, \vec{v}, t)$ :

$$\frac{\partial f_a}{\partial t} + \mathbf{v} \cdot \nabla f_a + \frac{\mathbf{F}_a}{m_a} \cdot \nabla_v f_a = \sum_{\beta} J_{a\beta} + \Gamma_a \quad (2.1)$$

where  $\nabla_v$  and  $\nabla$  are the gradient operators with respect to  $\vec{v}$  and  $\vec{x}$  variables,  $\mathbf{F}_\alpha = q_\alpha(\mathbf{E} + \mathbf{v}_\alpha \times \mathbf{B})$  is the Lorentz force the  $\alpha$ -th population is subject to,  $\sum_{\beta} J_{a\beta}$  stands for elastic collision source term between  $a$  and  $\beta$  particles, whereas  $\Gamma_a$  includes inelastic interactions. One of the many possible derivation of 2.1 is done by first building a continuity equation for  $f_a$ , keeping in mind that gradients act also on the velocity phase space, and eventually averaging what is left over a physical volume. Note that many different approach have been proposed for modeling the source term in 2.1, which indeed comes out of the average of fluctuating terms: the modeling approach chosen for this thesis will be introduced in section 2.3.1.

## 2.2 Derivation of Euler set of equations

One of the most important features of fluid regimes is the fact that collisions among particle play a key role in the definition of the physical model. When a particle  $\alpha$  travels for a distance before colliding with another particle  $\beta$  such that this distance  $\lambda_{\alpha\beta}$  is much smaller than a reference macroscopic length  $L_0$  of the problem, then the assumption of high-number of collision can be brought about in the model. The nondimensional coefficient used for this kind of analysis is the Knudsen number

$$Kn_{\alpha\beta} = \frac{\lambda_{\alpha\beta}}{L_0} \quad (2.2)$$

and for  $Kn_{\alpha\beta} < 0.01$  the above mentioned assumption holds: that being the case of Hall-Thrusters treated within this thesis project as far as electrons are concerned. High number of collisions means that the information exchange among different species across the domain happens almost instantaneously: this is the so called thermalization effect in which different species reach immediately thermal equilibrium when they collide with each other. Enforcing this assumption means knowing "for free" the shape of the distribution

function without having to compute it through equation 2.1, that is the full Maxwell-Boltzmann distribution:

$$f_{\alpha}^M(\mathbf{u}_{\alpha}) = n_{\alpha} \left( \frac{m_{\alpha}}{2\pi k_B T_{\alpha}} \right)^{\frac{3}{2}} \exp \left[ -\frac{m_{\alpha} (\mathbf{u}_{\alpha} - \bar{\mathbf{u}})^2}{2k_B T_{\alpha}} \right] \quad (2.3)$$

where  $\bar{\mathbf{u}}$  is the fluid average (or bulk) velocity,  $\mathbf{u}_{\alpha}$  is the particle velocity of the  $\alpha$ -th population,  $n_{\alpha}$  the number density,  $k_B$  the Boltzmann constant and  $T_{\alpha}$  the  $\alpha$ -th population temperature. It is therefore essential to understand how each hypothesis outline the feature of the physical model used. Within the limits of Hall-Thrusters it is further possible to assume that neither a relativistic approach nor a quantum approach have to be used. A further approximation can be made when defining the concept of single particle thermal energy  $\epsilon_k$  together with the one of potential energy related to electrostatic interaction  $\epsilon_{pot}$ . When in fact the ratio  $\frac{\epsilon_k}{\epsilon_{pot}}$  happens to be high enough it is said that Coloumb potential interactions are negligible. This ratio can be rewritten also as:

$$\frac{\epsilon_k}{\epsilon_{pot}} \approx \sqrt{(n\lambda_D^3)^3} \quad (2.4)$$

where  $(n_0\lambda_D^3)$  represents the number of particles within the Debye sphere, and  $\lambda_D$  is the Debye length (here below in Gauss notation), representing the maximum distance within which particles are said to still feel charge separation.

$$\lambda_D = \sqrt{\frac{k_B T}{4\pi n q_e^2}} \quad (2.5)$$

where  $nq_e^2$  is referred to the electrostatic interaction between charged particles, whereas  $k_B T$  is the particle thermal energy. Whenever this last hypothesis holds true the electrostatic potential is said not to influence transport properties of molecules, whose dynamics is dominated by their thermal agitation motion, and it is eventually possible to define a precise kinetic relationship between Pressure  $P$  and Temperature  $T$ , that is [11]:

$$P = nk_B T \left(1 - \frac{1}{24\pi n \lambda_D^3}\right) \approx nk_B T \quad (2.6)$$

which allows further to define the following concept of internal energy

$$e = \frac{3}{2}nk_B T - \frac{nT}{8\pi n \lambda_D^3} \approx \frac{3}{2}nk_B T \quad (2.7)$$

where  $e$  is the internal energy density. Both equations 2.6 and 2.7 will be used, together with all the above mentioned assumptions, to build a further simplification: describe and study plasma with fluid models. In order to achieve that, the following step is to build quantities that express the fluid flow field as a function of the phase space variables, and this is brought about through an ensemble average procedure:

$$\langle \psi(\vec{v}) \rangle_a = \frac{\int \psi f_a d^3\vec{v}}{\int f_a d^3\vec{v}} \quad (2.8)$$

where  $\psi$  is a statistical moment of the distribution function,  $f_a$  is the probability of finding one particle of species  $a$  at position  $\vec{x}$ , time  $t$  and at velocity  $\vec{u}_a$ , finally the integration procedure is done over the entire phase velocity space. At this point equation 2.3 undergoes the ensemble average procedure, resulting in equation 2.9.

$$\begin{aligned} \langle \frac{\partial f_\alpha}{\partial t}, \psi(\vec{v}) \rangle + \langle \mathbf{v} \cdot \nabla f_\alpha, \psi(\vec{v}) \rangle + \langle \frac{\mathbf{F}_\alpha}{m_\alpha} \cdot \nabla_v f_\alpha, \psi(\vec{v}) \rangle = \\ \langle \sum_\beta J_{\alpha\beta}, \psi(\vec{v}) \rangle + \langle \Gamma_\alpha, \psi(\vec{v}) \rangle \end{aligned} \quad (2.9)$$

where the operation  $\langle, \rangle$  is defined through the operator  $\Lambda$  in the following way:

$$\Lambda|_{\psi(\vec{v}) \in \mathbb{R}^m} : \vec{x}(\in \mathbb{R}^n) \rightarrow \mathbb{R}^n \otimes \mathbb{R}^m = \langle \psi(\vec{v}), \vec{x} \rangle = \int \psi(\vec{v}) \vec{x} d^3\vec{v} \quad (2.10)$$

---

<sup>2</sup>Notation simplified for:  $\int_{-\inf}^{+\inf} \int_{-\inf}^{+\inf} \int_{-\inf}^{+\inf} (...) dv_x dv_y dv_z$

To different values of  $\psi$  it can be associated a precised statistical moment: indeed taking all the infinite number of statistical moment it is possible to reconstruct the exact solution  $f_a$  of the kinetic equation 2.1. As far as the aim of this project is concerned, of interest will be just the analysis of the first 3 moments: this introduces, as it is possible to infer, an approximation of the exact model solution, nonetheless not so far from it being all the hypothesis valid in the operative range of Hall-Thrusters study. The first moment corresponds to the choice of  $\psi = 1$ , resulting thus in the continuity equation 2.11

$$\frac{\partial n_\alpha}{\partial t} + \nabla(n_\alpha \vec{v}) = \int C_\alpha d^3\vec{v} \quad (2.11)$$

where  $C_\alpha$  here stands for a simplified notation of the elastic and inelastic source terms together, whose detailed derivation will be presented in the following section. It is straightforward to notice that, multiplying both sides of 2.11, the mass density conservation equation is retrieved. The second moment is obtained by enforcing  $\psi = m_a \vec{v}$ , from which it is possible to define the average fluid velocity  $u_a$  as follows:

$$\langle m_a \vec{v} \rangle = m_a \vec{u}_a \quad (2.12)$$

where:

$$\vec{v}_a = \vec{u}_a + \vec{v}'_a \quad (2.13)$$

in which  $\vec{v}'_a$  stands for the disordered fluid velocity fluctuations of species a-th. Equation 2.1 eventually becomes the momentum density conservation equation:

$$\frac{\partial \rho_a \vec{u}_a}{\partial t} + \nabla \cdot (\rho_a \vec{u}_a \otimes \vec{u}_a) + \nabla \cdot (\rho_a \langle \vec{v}' \otimes \vec{v}' \rangle_a) = q_a n_a (\vec{E} + \vec{u}_a \times \vec{B}) + R_a \quad (2.14)$$

where  $R_a$  is the momentum source contribution of elastic and inelastic terms that once again will be treated with more detailed in the following section. To be noticed in equation 2.14 is the divergence of the fluctuations-tensor term, whose argument can be rewritten as follows:

$$\rho_a \langle \vec{v}' \otimes \vec{v}' \rangle_a = P_a \mathbf{I} + \mathbf{\Pi}_a \quad (2.15)$$

that is an isotropic contribution  $P_a \mathbf{I}$  and an an-isotropic one  $\mathbf{\Pi}_a$ . The second part would not be handled in this work, but according to the physics of the problem it is possible to formulate different hypothesis on how to give an analytical derivation: within the field of continuum mechanics, elasticity assumptions of the material implies the existence of a potential from which the an-isotropic part of the stress tensor is derived, indeed the hypothesis of Newtonian fluid allows to express it as a function of velocity gradients. When dealing with strong electromagnetic interaction between charged particle like electrons and ions this simplification happens to give very non accurate results, and since a throughout derivation of such tensor is out of the topic of this work, attention will be driven towards the isotropic part of 2.15. Assuming a thermal equilibrium the distribution function chosen to derive fluid variables is the Maxwell-Boltzmann one 2.3, that, thanks to its even property and to symmetry of the integration domain, allows the following simplification (in Einstein notation chosen a local Cartesian orthogonal system of reference):

$$P_{a,ij} = m_a \int v'_i v'_j f_a(v'^2) d^3 \vec{v} = P_{a,ii} = \frac{1}{3} \int m_a v'^2 f_a d^3 \vec{v} = n_a k_B T_a \quad (2.16)$$

in which in the last step, to every degrees of freedom is associated  $\frac{1}{2} k_B T$ . Note the analogy of the kinetic derivation of the Pressure term (elements on the diagonal of the isotropic tensor part of 2.15) and the Temperature, already verified in 2.6. The last equation that will be introduced as part of the set of equations within the multi-fluid derivation, is the one of total energy density  $\rho_a E_a = \rho_a \frac{1}{2} u_a^2 + \frac{P_a}{\gamma - 1}$  (indeed the contribution coming from kinetic and internal energy), corresponding to the third statistical moment and the choice of  $\psi = \frac{1}{2} m_a v^2$ . Noting that the electric field and not the

magnetic field is involved in the system energy transfer through the Joule effect and that analogous considerations seen in the derivation of equation 2.14 hold here, as far as the velocity decomposition and the energy term arisen from collisional interactions  $S_a = \int \frac{1}{2} m_a v^2 C_a d^3 \vec{v}$ . Equation 2.3 turns into the following:

$$\frac{\partial \rho_a \langle v^2 \rangle_a}{\partial t} + \nabla \cdot \left( \frac{1}{2} \rho_a \langle \vec{v} v^2 \rangle_a \right) = \vec{J}_a \cdot \vec{E} + S_a \quad (2.17)$$

where  $\vec{J}_a = q_a n_a \vec{u}_a$  is the current density generated by the a-th population. Thanks to the zero average value of linear combinations of velocity fluctuations, it is possible to introduce the following decomposition:

$$\langle \vec{v} v^2 \rangle = 2 \vec{u}_a \langle \vec{v}' v' \rangle_a + \langle \vec{v}' v'^2 \rangle_a + u_a^2 \vec{u}_a + \vec{u}_a \langle v'^2 \rangle_a \quad (2.18)$$

which eventually gives rise to the four following terms:

- $\frac{1}{2} \rho_a u_a^2 \vec{u}_a$  is the kinetic energy flux, transported by the ordered motion of particles
- $\frac{3}{2} k_B T_a \vec{u}_a$  (obtained using 2.20) is the thermal flux transported by the ordered motion of particle, that is the one responsible for heat convection
- $\rho_a \langle \vec{v}' v' \rangle_a \vec{u}_a = \mathbf{P}_a \cdot \vec{u}_a$  is the work associated to Pressure tensor stresses
- $\frac{1}{2} \rho_a \langle \vec{v}' v'^2 \rangle_a = \vec{Q}_a$  is the thermal flux transported by the disordered motion of particle, that is the one responsible for heat conduction.

Within the working frame of this thesis, no external heat flux due to conduction will be considered, forcing the adiabaticity to hold valid for each thermodynamic transformation that will characterize the presented test cases simulations.



## 2.3 Source terms derivation

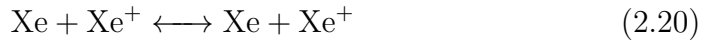
In the following sections a throughout derivation of the different source terms that complete the set of equations derived from the first three statistical moments will be presented. At first it will be presented an analytical derivation of the collisional source term, whereas in the second part a simplified set of Maxwell equations will close the electromagnetic source.

### 2.3.1 Collisional source terms

The type of collisions that will be handled within this thesis framework are the elastic collisions, ionization process, in which electrons ionize Xenon gas atoms giving birth to electrons and ions, and eventually excited state of Xenon gas atoms will be taken into account, but no recombination will be considered, due to the very short time ions stay within the reactor. The very first thing is to understand how to model the source term that in the previous derivation was mentioned as  $C_a$ , comprehensive of elastic and inelastic contributions  $\sum_{\beta} J_{\alpha\beta} + \Gamma_{\alpha}$ . In a final step the concept of collisional frequency is first introduced, then properly derived ( Appendix .1).

#### 2.3.1.1 Elastic collisions

The main elastic collisional events that will be considered are: the elastic scattering of electrons interacting with neutrals 2.19 and the elastic scattering of ions interacting with neutrals 2.20.



An exact analytical expression for the elastic collisional source term exists: the Boltzmann integral, whose solution, despite being analytically correct

and therefore precise, is very cumbersome to obtain computationally. Therefore many are the approaches proposed in literature to model the elastic collisions source term [3] in an alternative way, one of the most used is indeed the BGK or Relaxation-Time approximation, which bounds this source term to the inverse of the relaxation time: indeed very long relaxation time means that collisional phenomena aimed at bringing the system back at thermodynamic equilibrium are negligible. Collisional terms can be also neglected when system reaches thermodynamic equilibrium since everything is then static and velocity distribution is therefore isotropic. Given those two facts, the elastic part of term  $C_a$  can be written as:

$$C_{a,elastic} = -\frac{f_a - \mathcal{M}}{t_r} \quad (2.21)$$

where  $\mathcal{M}$  stands for the Maxwellian distribution, to which  $f_a$  tends in case of thermodynamic equilibrium, whereas  $t_r$  is the relaxation time. Within the assumptions considered here, post-collisional electrons' velocities follow a Maxwellian distribution (isotropic scattering) for which the average (post-collisional) velocity  $\bar{\mathbf{u}}$  is zero. Neutral gas atoms are moreover assumed to be still (typical injected velocity of Xenon gas is around 300 m/s, electrons velocities are of the order of  $10^6$  m/s, whereas ions near exhaust can reach  $10^4$  m/s [8]), saving the only contribution of electrons velocities inside the integration process.

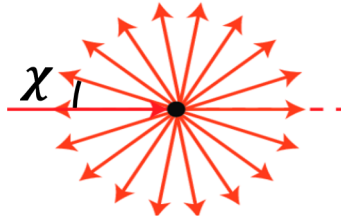


Figure 2.1: Isotropic scattering in case of still neutrals

The relaxation time  $t_r$  is modeled through its inverse:

$$t_r = \frac{1}{\nu} \quad (2.22)$$

where  $\nu$  is a frequency variable also called *collisional frequency*, whose derivation will be proposed in this section. As far as **collisional elastic source term for continuity equation** is concerned 2.11, when integrating all over the velocity phase space, the distribution function  $f_a$  will relax to a Maxwellian one, and therefore:

$$\begin{aligned} \int C_{a,elastic} d^3\vec{v} &= -\frac{1}{t_r} \int (f_a - \mathcal{M}) d^3\vec{v} \\ &= -\frac{1}{t_r} (\langle f_a \rangle - \mathcal{M}) \\ &= -\frac{1}{t_r} (\mathcal{M} - \mathcal{M}) = 0 \quad (2.23) \end{aligned}$$

Moving to the **elastic part of momentum equation source term**  $R_a$  in equation 2.14, the term surviving the ensemble average is only the one connected to  $f_a$ , since due to isotropic scattering the post-collisional average velocity is void:

$$\begin{aligned} R_{a,elastic} &= \int m_a \vec{v} C_{a,elastic} d^3\vec{v} = \\ &= - \int m_a \vec{v} \frac{1}{t_r} (f_a - \mathcal{M}) d^3\vec{v} = -\frac{\rho_a \vec{u}_a}{t_r} \\ &= -\nu_{el,a} \rho_a \vec{u}_a \quad (2.24) \end{aligned}$$

where  $\nu_{el,a}$  is the collisional frequency connected to elastic scattering between population  $a$  and neutral gas atoms. For both ions and electrons population colliding with neutrals a change in momentum is proportional to each particle momentum at the moment in which they interact elastically with each other. To be noted is the fact that for very long relaxation time changes in momentum are negligible. Eventually also the energetic contribution from equation 2.17 can be considered negligible for both electrons and ions population when elastically colliding with neutrals. This is of course a big approximation in case no electromagnetic interaction has to be accounted for: it can be shown in fact that energy source term connected to Joule heating is orders of magnitude higher than those connected for example to the

energy exchange in elastic collisions between neutral gas and charged ions. On the other hand, thanks to the post-collisional isotropic scattering and the fact that electrons mass is five orders of magnitude smaller than that of neutrals, the exchange of energy between electrons and neutrals gas atoms can be neglected:

$$\frac{\Delta\epsilon}{\epsilon} = 2\frac{m_e}{M_n}(1 - \cos\chi) \approx 0 \quad (2.25)$$

where  $m_e$  is the electron mass, whereas  $M_n$  is the neutral gas (Xenon) mass, and  $\chi$  is the isotropic scattering angle of Fig. 2.1. To conclude hence the **elastic part of the source term connected to energy equation** is void:

$$S_{a,elastic} = \int \frac{1}{2}m_a v^2 C_{a,elastic} d^3\vec{v} = - \int \frac{1}{2}m_a v^2 \frac{1}{t_r} (f_a - \mathcal{M}) d^3\vec{v} = 0 \quad (2.26)$$

### 2.3.1.2 Inelastic collisions

The only two inelastic collision phenomena that will be considered within this thesis project are the ionization process, in which one electron ionizes one atom of Xenon giving birth to another electron and a Xenon ion 2.27, and excitation, in which electrons may excite neutral atoms 2.28, thus losing energy [4].



For both electrons and ions population the inelastic collisional source connected to 2.11 has to do with ionization only: from 2.27 it is possible to see that when impacting against neutrals electrons may give birth to another electron and an ion. The frequency with which this phenomenon happens is represented by the ionization frequency  $\nu_{iz}$ , whose analytical derivation will

be treated when introducing the reaction rates in section 2.3.1.3. For electrons ( $e$ ) and ions species ( $i$ ) the **inelastic part of source term connected to continuity equation** are respectively:

$$\int C_{e,inelastic} d^3\vec{v} = \rho_e \nu_{iz} \quad (2.29)$$

$$\int C_{i,inelastic} d^3\vec{v} = m_i n_e \nu_{iz} \quad (2.30)$$

Ionization affects instead only ions when dealing with momentum: indeed every time an ion is generated out of a neutral gas atom, the former indeed contributes to the ions-momentum 2.14 : this is why to pre-multiply the electrons number density responsible for the ionization process is the ions mass and not the electrons mass. For what concerns the **inelastic part of the source term connected to momentum equation** only a term connected to ions population survives:

$$R_{i,inelastic} = \int m_i \vec{v} C_{i,inelastic} d^3\vec{v} = m_n n_e \nu_{iz} \quad (2.31)$$

where the subscript  $i$  stands for ions and  $n$  for neutrals. No net momentum contribution from electrons population is considered since, as already mentioned, post collisional electrons velocities distribute according to a Maxwellian distribution centered around the zero value. The most elaborated contribution of ionization, and the only one of the excitation term, appears in the total energy conservation equation. On one side electrons population energy density, after a ionization process, tends to decrease: the reason is that the same amount of energy owned before just by one electron (ionization energy) now has to be split between two electrons (post-collisional with zero average velocity). Whereas as far as excitation is concerned the reasoning is much more straightforward: the energy given to excite neutrals is simply just lost by electrons, and the frequency with which this phenomenon happens is encapsulated in  $\nu_{ex}$ , whose derivation will be carried out in next section. For what concerns the **inelastic part of the source term connected to energy equation** for ions population it holds:

$$S_{e,inelastic} = \int \frac{1}{2} m_e v^2 C_{e,inelastic} d^3 \vec{v} = -\epsilon_{iz} n_e \nu_{iz} - \epsilon_{ex} n_e \nu_{ex} \quad (2.32)$$

where  $S_{e,inelastic} = \int \frac{1}{2} m_e v^2 C_{e,inelastic} d^3 \vec{v}$ , subscript  $e$  refers to electrons, whereas  $\epsilon_{iz}$  and  $\epsilon_{ex}$  are respectively the first ionization energy ( $\approx 12.13$  [eV]) and the activation energy (chosen in this work to be 12 [eV]). The logic behind the derivation of the energetic terms of ions follows from that done already for the momentum term: indeed in the moment an ion is generated from a Xenon atom it runs into the system with the same energetic features of this latter one:

$$S_{i,inelastic} = \int \frac{1}{2} m_i v^2 C_{i,inelastic} d^3 \vec{v} = m_n n_e \nu_{iz} \left( \frac{1}{2} u_n^2 + \frac{3}{2} k_B T_n \right) \quad (2.33)$$

where subscript  $n$  refers to neutrals.

### 2.3.1.3 Reaction rates

The conclusive part of the derivation of collisional source terms has to do with the analytical formulation of the relaxation time introduced at the beginning of the section. Nonetheless usually to be modeled is the inverse of the relaxation time, that is the frequency with which interactions, both elastic and inelastic, happen among the system particles. A very first approximation, called thermal approximation, can be to derive the collisional frequency as follows:

$$\nu \approx n_n \bar{\sigma} v_{th}(T_a) \quad (2.34)$$

where  $n_n$  is the background neutrals number density,  $\bar{\sigma}$  is the average scattering cross section (for the following analysis it will assume the constant value of  $10^{-19} \text{ m}^2$ ) and  $v_{th}(T_a) = \sqrt{\frac{8T_a k_B}{\pi m_a}}$  is the  $a$ -th species thermal velocity.

One of the aim of this work is to try to obtain a more precise and literature based derivation of the collision frequency. Since particle do not have a single velocity, but as seen in the kinetic equation derivation, can assume a distribution of velocity within the phase space, a more accurate formulation can start from the following expression [16]:

$$\nu = n_n \int \int \int \sigma(\vec{v}) \vec{v} f(\vec{v}) d^3 \vec{v} = n_n k(T_a) \quad (2.35)$$

where now  $\sigma$  is not a constant anymore but a function of velocity phase space,  $f(\vec{v})$  contains the information on how particle velocities are distributed and  $k(T_a)$  is called **reaction rate**. Focusing on the reaction rate term, integrating over all phase space helps taking into account all particle velocities contributions. The distribution function considered within this frame of work will be the Maxwell-Boltzmann one 2.3. Equation 2.35 can be further developed by introducing the kinetic energy term  $E = \frac{1}{2} m_e v^2$ , which eventually returns the following expression for the reaction rate:

$$k(T_a) = \left( \frac{1}{\pi m_e} \right)^{1/2} \left( \frac{2}{k_B T_a} \right)^{3/2} \int_0^\infty \sigma(E) E e^{-\frac{E}{k_B T_a}} dE \quad (2.36)$$

## 2.3.2 Electromagnetic field

Typically in Hall-Thrusters a voltage drop of around 300 Volt [8] is set between cathode and anode: this allows electrons to transit within the engine, get captured by radial magnetic field and ionize Xenon gas atoms. All this mechanism generates a self-consistent behavior of the plasma: different density distribution of charged particles generates an electric field, which accelerates them and creates eventually a self-consistent magnetic field. The created electric field thus, generated right where the electron mobility is at lowest, accelerates ions outside the engine. Within this thesis framework no self-consistent magnetic field will be taken into account, allowing essentially a strong simplification of the set of Maxwell equations needed to close the set of fluid-dynamic equations. The only unknown left to be defined is in fact the electric field, for which the Poisson equation will be used: here below in

Gauss notation.

$$\Delta\phi = -\rho_Q \quad (2.37)$$

where  $\Delta$  is the Laplace operator, whereas  $\rho_Q = \sum_i^{NS} q_i n_i$  is the plasma charge density and  $\phi$  is the electrostatic potential. Whereas magnetic field will appear in both components of momentum equation 2.14 due to its peculiar an-isotropic properties, electric field will appear instead only on the x-component of momentum equation and indeed it will be the main source of energy as far as total energy density of electrons and ions are concerned.

### 2.3.3 Full set of dimensional equations

The final set of equations comprehends thus 2 scalar equations (continuity and energy conservation) and 1 vectorial equation (Momentum with x and y or azimuthal component) for each species, plus an extra equation for the self-consistent electric field: here below in SI notation.

$$\frac{\partial \rho_e}{\partial t} + \nabla(\rho_e \vec{v}) = \rho_e \nu_{iz} \quad (2.38)$$

$$\frac{\partial \rho_e \vec{u}_e}{\partial t} + \nabla \cdot (\rho_e \vec{u}_e \otimes \vec{u}_e + P) = q_e n_e (\vec{E} + \vec{u}_a \times \vec{B}) - \rho_e \vec{u}_e \nu_{el} \quad (2.39)$$

$$\frac{\partial \rho_e E}{\partial t} + \nabla \cdot (\rho_e E \vec{u}_e + P \vec{u}_e) = \vec{E} \cdot \vec{J}_e - \epsilon_{iz} n_e \nu_{iz} - \epsilon_{ex} n_e \nu_{ex} \quad (2.40)$$

$$\frac{\partial \rho_i}{\partial t} + \nabla(\rho_i \vec{v}) = m_i n_e \nu_{iz} \quad (2.41)$$

$$\frac{\partial \rho_i \vec{u}_i}{\partial t} + \nabla \cdot (\rho_i \vec{u}_i \otimes \vec{u}_i + P) = q_i n_i (\vec{E} + \vec{u}_i \times \vec{B}) - \rho_i \vec{u}_i \nu_{il} + m_n n_e \nu_{iz} \vec{u}_n \quad (2.42)$$



$$\frac{\partial \rho_i E}{\partial t} + \nabla \cdot (\rho_i E \vec{u}_i + P \vec{u}_i) = \vec{E} \cdot \vec{J}_i + m_n n_e \nu_{iz} \left( \frac{1}{2} u_n^2 + \frac{3}{2} k_B T_n \right) \quad (2.43)$$

$$\Delta \phi = -\frac{\rho Q}{\epsilon_0} \quad (2.44)$$

No equations for neutral gas atoms are included, as mentioned in the introductory section of this project, since neutral gas atoms is assumed to be still and with a constant density profile in background.

## 2.4 Nondimensionalization

A further feature of the presented model and indeed of the developed software, is the possibility to run nondimensional simulations. The choice to develop a nondimensional analysis was in fact more motivated by the need, especially in implicit simulations, to reduce the numerical error due to round-off of either very large or very small numbers arithmetic, rather than study the physical behavior of plasma by interpreting values of nondimensional coefficients. Every physical unknown of the problem was then split into a reference variable(  $^*$  ) and its nondimensional counterpart (  $^-$  ) for every a-th species:

$$\begin{aligned} u &= u^* \bar{u} \\ P &= P^* \bar{P} \\ v &= v^* \bar{v} \\ E &= E^* \bar{E} \\ \rho &= \rho^* \bar{\rho} \end{aligned}$$

Not all reference variables are of free choice, since nondimensional coefficients for convective fluxes (LHS of equations 2.38-2.43) are set to one in order to not interfere with the eigenvalue structure of the Euler set of equations, needed in order to compute numerical fluxes. The choice of reference values left out from the just mentioned constrictions is of paramount importance

especially within implicit schemes: it will not always be possible in fact to set to unity all unknowns, therefore, in cases like the choice of reference velocities for electrons and ions, a fine tuning of which reference values works better than the other one is essential in order to improve in a consistent way the convergence properties of the method. More details will be given in section 3.6.1.2.

# Chapter 3

## Numerical tools

In the following section main numerical tools to develop the software *FORMICA* will be introduced. At first it will be presented the classical Finite Volume Method for 1D grid, with a quick overview on boundary conditions and temporal discretization. In a second frame Poisson's solver algorithm will be approached together with the possibility to non-uniformly discretize domain's mesh. Last part will be devoted to various acceleration technique.

### 3.1 Finite Volume Method for 1D problem

During the last decades, the Finite Volume method has become one of the most employed technique for simulating a wide variety of flows governed by hyperbolic equations [17]. The basic idea of this method is to subdivide the computational domain  $\Omega = \sum_i \Omega_i$  into a disjoint set of finite cells or volumes and to apply inside each cell the conservation laws. Within this thesis framework the domain 1D, therefore mesh is composed by one-dimensional cells as shown in Fig. 3.1. Solution is stored at the center of each cell. The algorithm discretizes the system of PDE's written in an integral form:

$$\int_{\Omega_i} \frac{\partial Q}{\partial t} d\Omega + \int_{\Omega_i} \nabla \cdot \vec{F} d\Omega - \int_{\Omega_i} \mathbf{S} d\Omega = 0 \quad (3.1)$$

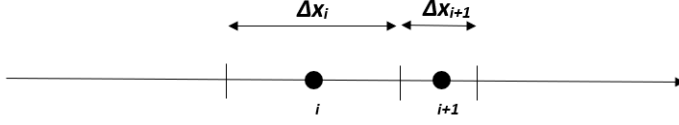


Figure 3.1: 1D Mesh discretization

that, through divergence theorem becomes:

$$\int_{\Omega_i} \frac{\partial Q}{\partial t} d\Omega + \sum_{i \pm \frac{1}{2}} \vec{F}_i \cdot \vec{n}_i - \int_{\Omega_i} \mathbf{S} d\Omega = 0 \quad (3.2)$$

where indeed  $i \pm \frac{1}{2}$  are the domain interfaces that separate cells from each others, and  $Q$  the conservative variables. Applying the mean integral value theorem to the integrals left to be discretized in 3.2, for each cell  $i$  equation 3.2 reduces to:

$$\frac{\partial \bar{Q}_i}{\partial t} + \frac{1}{\Delta x_i} (F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}) = \bar{S}_i \quad (3.3)$$

where  $\bar{Q}$  stands for the mean value conservative variables assume all over the  $i$ -th cell, and that is then used in the numerical scheme to reconstruct values at the interfaces.

## 3.2 Boundary conditions

Boundary conditions can be forced both for the fluid-dynamics set of equation and the Poisson solver linear system as it will be shown in section 3.5.

In the first case boundary conditions are enforced in a strong way through the computation of fluxes in the extreme ghost cells: indeed primitive variables  $(\rho, T, \vec{u})$  can be imposed in the ghost cells: it will be then the numerical scheme, by sorting the eigenvalues of the boundary cells, to decide whether the information goes out of the boundary (subsonic regime) into the domain, or goes out of the domain (supersonic regime) making hence void the imposed value at the ghost cells. Within the software there is the possibility to enforce

Dirichlet and Neumann boundary conditions (only homogeneous): this latter to avoid noisy boundary conditions that might create too much numerical instabilities at the boundaries of the domain.

As far as the Poisson's solver is concerned both strong Dirichlet and Neumann boundary conditions as well as periodic ones can be enforced: this mainly to be able to retrieve some of essential results within validation tests for the Poisson solver itself.

### 3.3 Numerical fluxes

The numerical scheme for fluxes computation used within this thesis framework is the HLLC scheme [22], a modification of the HLL scheme. Given  $S^* = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)}$  and according to the Pressure-Velocity based wave speed estimation the following fluxes structure is obtained:

$$\begin{aligned} \mathbf{F}_{*L} &= \mathbf{F}_L + S_L (\mathbf{Q}_{*L} - \mathbf{Q}_L) \\ \mathbf{F}_{*R} &= \mathbf{F}_{*L} + S_* (\mathbf{Q}_{*R} - \mathbf{Q}_{*L}) \\ \mathbf{F}_{*R} &= \mathbf{F}_R + S_R (\mathbf{Q}_{*R} - \mathbf{Q}_R) \end{aligned} \quad (3.4)$$

By moreover imposing the following conditions on the approximate Riemann solver

$$\begin{aligned} u_{*L} &= u_{*R} = u_* \\ p_{*L} &= p_{*R} = p_* \\ v_{*L} &= v_L, v_{*R} = v_R \end{aligned} \quad (3.5)$$

equation 3.4 can be reworked in order to deliver the solution vectors for conservative variables across the *StarRegion*:

$$\mathbf{Q}_{*K} = \rho_K \left( \frac{S_K - u_K}{S_K - S_*} \right) \begin{bmatrix} 1 \\ S_* \\ v_K \\ \frac{E_K}{\rho_K} + (S_* - u_K) \left[ S_* + \frac{p_K}{\rho_K (S_K - u_K)} \right] \end{bmatrix} \quad (3.6)$$

The final HLLC fluxes scheme can be eventually written as:

$$\mathbf{F}_{i+\frac{1}{2}}^{hllc} = \begin{cases} \mathbf{F}_L & \text{if } 0 \leq S_L \\ \mathbf{F}_{*L} = \mathbf{F}_L + S_L (\mathbf{Q}_{*L} - \mathbf{Q}_L) & \text{if } S_L \leq 0 \leq S_* \\ \mathbf{F}_{*R} = \mathbf{F}_R + S_R (\mathbf{Q}_{*R} - \mathbf{Q}_R) & \text{if } S_* \leq 0 \leq S_R \\ \mathbf{F}_R & \text{if } 0 \geq S_R \end{cases} \quad (3.7)$$

### 3.4 Temporal discretization

Two main different temporal discretization schemes will be proposed within this thesis framework: the *Midpoint Euler scheme* and the *Backward Euler scheme*<sup>1</sup>.

#### 3.4.1 Midpoint Euler

Midpoint Euler (**ME**) is an explicit scheme, that works in an analogous way to the classical Forward Euler (FE) explicit scheme: residual is computed at a previous timestep and solution can be updated without having to solve any implicit computation. One of the feature Midpoint Euler method gains over the simpler explicit schemes is the fact that fluxes are updated at an intermediate timestep, through a simple Forward Euler iteration, in order to improve the precision of the prediction that in the Forward Euler scheme was merely depending on the derivative of the starting point.

$$\frac{\overline{\mathbf{Q}}_i^{n+1} - \overline{\mathbf{Q}}_i^n}{\Delta t} = \frac{1}{\Delta x} \left( \mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right) + \overline{\mathbf{S}}_i^n \quad (3.8)$$

Despite this scheme unavoidably implies a longer computational time, since fluxes depend on conservative variables whose values have to be computed at an intermediate step as well, Midpoint Euler is a much more numerically stable scheme than Forward Euler and is much less dissipating scheme

---

<sup>1</sup>Used exclusively within a Dual Time Stepping scheme which will be introduced in the following section

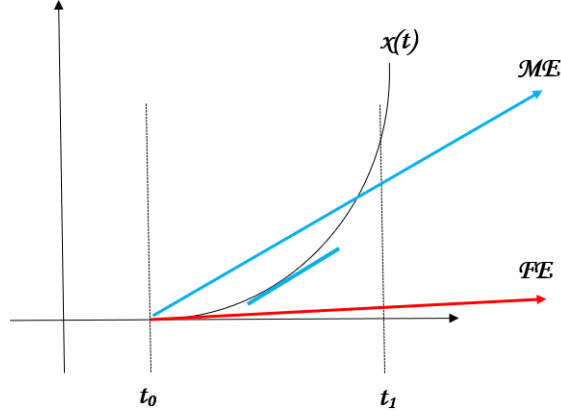


Figure 3.2: Midpoint Euler vs Forward Euler scheme

than the former one [10], that is pressure is much less likely drop quickly to zero values. Being an explicit scheme the stability criteria enforced by the Courant-Friedrich-Lewy (CFL) number has to be valid, that is  $CFL < 1$ , where

$$CFL = \frac{\Delta t}{\Delta x}(|u| + a) \quad (3.9)$$

being  $a$  the gas speed of sound.

### 3.4.2 Backward Euler

Inside the software *FORMICA* this implicit method is not provided as pure alternative to the Midpoint Euler scheme, but it will be used within another numerical method whose features will be presented in the following sections. Backward Euler (*BE*) scheme differs from the explicit ones mentioned in previous sections since at every timestep residual needs to be known at the next timestep in order to update the solution.

$$\frac{\overline{Q}_i^{n+1} - \overline{Q}_i^n}{\Delta t} = \frac{1}{\Delta x} \left( \mathbf{F}_{i-\frac{1}{2}}^{n+1} - \mathbf{F}_{i+\frac{1}{2}}^{n+1} \right) + \overline{S}_i^{n+1} = -\frac{\mathbf{R}_i^{n+1}}{\Delta x} \quad (3.10)$$

This being not always feasible and most of the time implying cumbersome computational operations, an alternative approach can be to use instead a linearized implicit scheme. Implicit residual is approximated with its first order Taylor expansion, so that finding the non-linear solution means using a Newton method that always stops at the very first iteration.

$$\mathbf{R}_i(\mathbf{Q}^{n+1}) = \mathbf{R}_i(\mathbf{Q}^n) + \frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial t} \Delta t^n = \mathbf{R}_i(\mathbf{Q}^n) + \frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_j^n} \Delta \mathbf{Q}_j^n \quad (3.11)$$

where  $\mathbf{R}_i(t^n)$  is the residual<sup>2</sup> and subscript  $i$  and  $j$  denotes the domain mesh cells. Using 3.11 as rhs of equation 3.8, dividing and multiplying by  $\Delta x$  and using the following notation  $\Delta \mathbf{Q}_i^n = \overline{\mathbf{Q}_i^{n+1}} - \overline{\mathbf{Q}_i^n}$ , it is possible to obtain the final shape of the linearized Backward Euler:

$$\left( \frac{\Delta x_i}{\Delta t^n} \mathbf{I} + \frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_j^n} \right) \Delta \mathbf{Q}_j^n = -\mathbf{R}_i(\mathbf{Q}^n) \quad (3.12)$$

where  $\frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_j^n}$  is the Jacobian double tensor that contains the residual derivative with respect to conservative variables. The more accurate is the shape of the Jacobian tensor, the better the implicit method is performing; this statement should not be underestimated though, since most of the time an analytical formulation is not available and the numerical approximation requires a non trivial formulation. Having said that the method is still a linearized one, meaning that indeed a higher timestep  $\Delta t$  than the explicit method is allowed ( $CFL \geq 1$ ), but too wide timestep can strongly compromise both the quality of physical solution and convergence stability in very unsteady simulation. The Jacobian tensor  $\frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_j^n}$  can be rewritten as:

$$\frac{\partial \mathbf{R}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_i^n} = \left( \frac{\partial \mathbf{F}_{i+\frac{1}{2}}(\mathbf{Q}^n)}{\partial \mathbf{Q}_i^n} - \frac{\partial \mathbf{F}_{i-\frac{1}{2}}(\mathbf{Q}^n)}{\partial \mathbf{Q}_i^n} \right) - \frac{\partial \mathbf{S}_i(\mathbf{Q}^n)}{\partial \mathbf{Q}_i^n} dx_i \quad (3.13)$$

where it can be easily seen that fluxes at interfaces depend on the values in the cells' nodes at both sides of the interface itself<sup>3</sup>. Numerical fluxes

---

<sup>2</sup> $\left( \mathbf{F}_{i-\frac{1}{2}}^n - \mathbf{F}_{i+\frac{1}{2}}^n \right) + \overline{\mathbf{S}}_i^n \Delta x$

<sup>3</sup>within this thesis framework a constant reconstruction (1st order scheme) has been proposed



Jacobian is preformed numerically by perturbing the conservative variable at the  $i$ -th cell:

$$\frac{\partial \mathbf{F}_{i+\frac{1}{2}}(Q_i, Q_{i+1})}{\partial Q_i} = \frac{\tilde{\mathbf{F}}(Q_i + \delta Q_i, Q_{i+1}) - \tilde{\mathbf{F}}(Q_i, Q_{i+1})}{\delta Q_i} \quad (3.14)$$

where  $\tilde{\phantom{x}}$  stand for numerical fluxes (and not physical ones), and  $\delta Q = 10^{-4} \max(|Q_i|, \text{typ}(Q_i) \text{sign}(Q_i))$  is the numerical perturbation of the conservative variable [12]. From equation 3.13 it is possible to see that the implicit operator has the shape of a block tridiagonal matrix. As far as the source term part of the Jacobian tensor is concerned, there are only block diagonal elements: more precisely the source term connected to collisions is computed numerically and it is fully implicit. The only part whose derivation is not fully nonlinear is the one connected to the electromagnetic source: indeed a full implicit derivation requires to compute for each cell a linear system as big as the square of the number of mesh's cells through the Poisson's solver, resulting in a computationally demanding effort by the software. The main strategy to overcome this issue is to simply take the electric field value as constant. The term in round brackets at the lhs of equation 3.12 is therefore a block-tridiagonal matrix (see Fig. 3.3), in which there will be as many block as the number of cells and each block will be  $(N_{eq}) \times (N_{cons})$ . Being

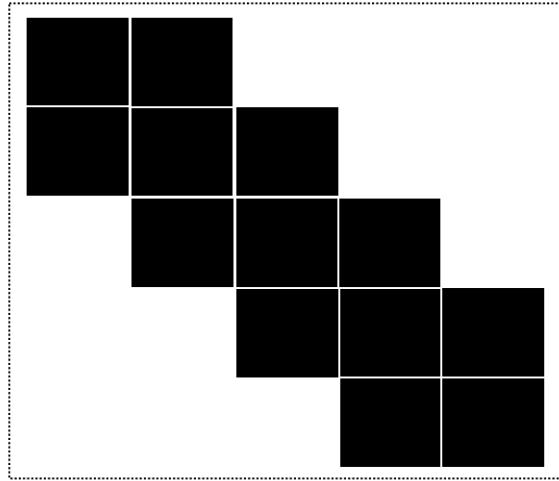


Figure 3.3: Block tridiagonal matrix

hence the Jacobian tensor a sparse matrix, being the size of the matrix proportional to the number of cells and not existing an efficient direct-method algorithm to solve a block tridiagonal matrix, the choice of the library for solving the linear system fell on the LIS library [20]. LIS (see Appendix .2) routines allow to store blocks of elements within a sparse matrix and solves linear system through iterative methods: in this thesis project GMRES solver with ILU preconditioner were chosen. This choice, compared to other direct method solver library, strongly reduces the amount of memory required to store all the information, leaving free space for the operative system cache and enabling hence a considerable boost within the computational operations timing. The "quality" of the Jacobian matrix strongly influences convergence properties of the implicit method: indeed the better the derivative is computed the better it will estimate with good precision the evolution of the solution in time. For this reason a further numerical tool has been introduced, that is a *predictor corrector* for the electric field  $\mathbf{E}$ . The method has been proposed by ([14]) and deals with a prediction of the electric field at timestep  $n + 1$ , by using density solution and electric field values itself from two past timesteps. More details will be given in the appendix (see Appendix .3). What further matters, as far as "quality" of the Jacobian matrix is concerned, is indeed the numerical approximation: it was in fact not feasible to run implicit simulation without a proper nondimensionalisation, that strongly reduced the numerical approximation error within the linear system solver, assuring a much more performing convergence property of the method.

### 3.5 Poisson's solver equation

In order to close the set of fluid-dynamics unknowns a further equation is needed in order to define the self-consistent electric field arising from the interaction between charged particles like ions and electrons. Not considering electrodynamics effects (magnetic field is only external and constant in space and time) one simple way to compute self-consistent electric field is by using the electrostatic potential  $\phi$ , which through the Poisson equation reads as

follows (in SI notation):

$$\Delta\phi = -\frac{\sum_i^{NS} q_i n_i}{\epsilon_0} = -\frac{\rho_Q}{\epsilon_0} \quad (3.15)$$

which in a 1D framework simplifies into:

$$\frac{\partial^2 \phi}{\partial x^2} = -\frac{\sum_i^{NS} q_i n_i}{\epsilon_0} = -\frac{\rho_Q}{\epsilon_0} \quad (3.16)$$

with NS the number of species. The numerical scheme used to discretize equation 3.16 is *Finite Difference* scheme (FD) and the domain discretization is carried out on cells' interfaces. The reason behind this choice is mainly due to the fact that a voltage drop in a Hall-Thruster engine implies a prescribed voltage for the cathode and a prescribed voltage for the anode as well, and hence Dirichlet boundary conditions have to be enforced at both extremes (the first and last interfaces) of the domain. Electrostatic potential is hence computed on cells' interfaces (see equation 3.17) allowing moreover a straight forward derivation of the electric field at every cell's internal node through a centred-difference numerical scheme. Before introducing the numerical algo-

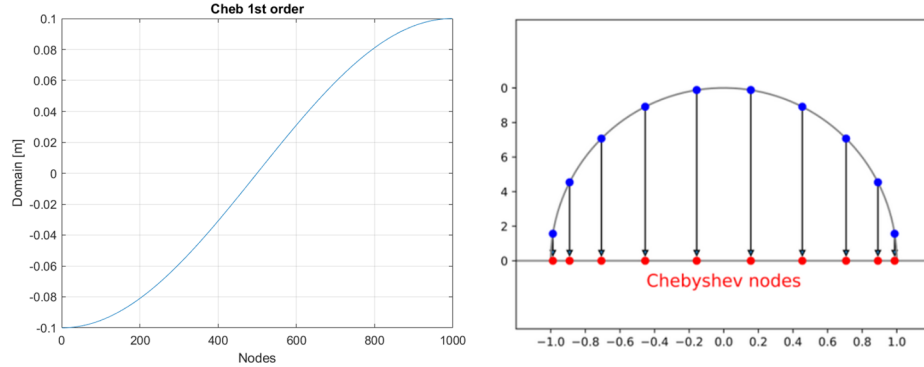


Figure 3.4: Chebyshev nodes

gorithm used to discretize 3.16, a further feature of the *FORMICA* software is introduced here, that is the possibility to run explicit and implicit simulation using a full non-homogenous 1D grid. For the sake of simplicity *Chebyshev* nodes of 1st order were chosen in order to define the central nodes of the

cells: nodes were distributed hence according to a cosinusoidal function, and the final result is shown in Fig.3.4. Having said that the FD algorithm reads as follows [15]:

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{i+\frac{1}{2}} = \frac{2}{\Delta x_{i+\frac{3}{2}} + \Delta x_{i+\frac{1}{2}}} \left[ \frac{\phi_{i+\frac{3}{2}} - \phi_{i+\frac{1}{2}}}{\Delta x_{i+\frac{3}{2}}} - \frac{\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}} \right] \quad (3.17)$$

Thanks to its tridiagonal shape, the linear system obtained by 3.16, can be efficiently resolved by the ad-hoc routines of the most performing library for direct-method solver, LAPACK [2]. Despite in fact being the matrix in 3.16 as big as the number of cells squared, LAPACK routine to solve tridiagonal system by means of Thomas algorithm, outperforms by large performances of any other library's utilities (see Appendix .4). Once obtained the electrostatic potential on each interface, a simple centered-difference numerical scheme is used to retrieve the value of the electric field at the cells' nodes; indeed

$$\vec{E} = -\frac{\partial \phi}{\partial \vec{x}} \quad (3.18)$$

reads as

$$E_i = -\frac{\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}}}{\Delta x} \quad (3.19)$$

## 3.6 Acceleration techniques

One of the most troublesome issues of developing a multi-fluid model (and software) is to properly reproduce the physics when typical flow times of the species at stack are orders of magnitude different. In plasma physics understanding the interactions between electrons and ions population is essential if one wants to investigate phenomena like self-consistent electric field. This means that numerical features of the code need to cope with features of the species with the shortest typical time: in this thesis electrons. This unavoidably requires long computational timing to observe the evolution of

ions dynamics, whose behavior defines the characteristics of an Hall-Thruster engine. In the following sections acceleration techniques will be introduced in order to speed up the computation without losing important information from electrons population. A Dual-Time-Stepping scheme strategy [23] is proposed, in which additional speed-up features like local timestepping and an algorithm for sub-iterations restart will be also presented.

### 3.6.1 Dual-Time-Stepping

One numerical method used in literature to reduce computational timings of *unsteady flow*<sup>4</sup> simulations is the *Dual-Time-Stepping* technique. One way

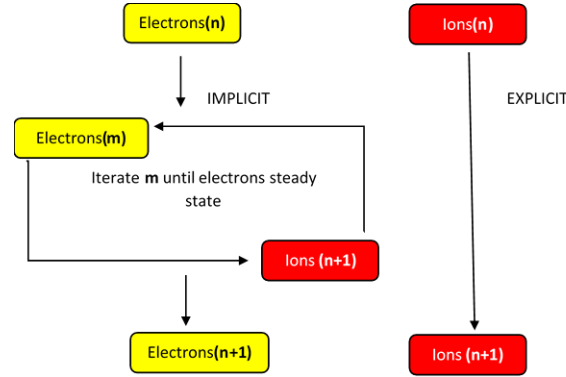


Figure 3.5: Electrons-Ions in Dual-Time-Stepping

to curb the issue of a huge gap between ions and electrons characteristic times is summed up in Fig.3.5 : first explicitly iterate ions, then keep them fixed and let electrons evolve until their dynamics stabilize around a sort of time-local steady state solution. The core of this method is in fact the introduction of a pseudo-time  $\tau$  that will have to deal with this phase of adaption of the electrons dynamics to the ions solution at previous physical timestep. Equation 3.3 hence is rewritten as follows:

$$\frac{\partial \bar{Q}_i}{\partial \tau} + \left( \frac{\partial \bar{Q}_i}{\partial t} + \frac{1}{\Delta x_i} (F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}) - \bar{S}_i \right) = 0 \quad (3.20)$$

<sup>4</sup>In this thesis project it was given priority to the possibility of reproducing transient phenomena, rather than just obtaining steady state results

where now the new residual  $\vec{R}^*$  is the term in round brackets (scaled by  $\Delta x$ ) of 3.20, so that equation 3.20 can be rewritten as:

$$\Delta x_i \frac{\partial \vec{Q}_i^*}{\partial \tau} = -\vec{R}^*(\vec{Q}^*) \quad (3.21)$$

where now  $\vec{Q}^*$  is the steady state solution of the pseudo-time  $\tau$  and the solution at the  $n + 1$  timestep of the physical-time  $t$ . All the advantages of implicit methods [5] can be exploited now to solve 3.21, since the goal is the electrons steady-state solution. By discretizing w.r.t.  $\tau$ :

$$\vec{R}^*(\vec{Q}^*)^{m+1} = \vec{R}^*(\vec{Q}^*)^m + \frac{\partial \vec{R}^*}{\partial \vec{Q}^*} \delta \vec{Q}^* \quad (3.22)$$

with  $m$  defined as the *sub-iteration index*, that is the pseudo-timestep. Introducing eventually:

$$\frac{\partial \vec{R}^*}{\partial \vec{Q}^*} = \left( \frac{\partial \vec{R}}{\partial \vec{Q}^*} (\vec{Q}^*)^m + \frac{\Delta x_i}{\Delta t} \right) \quad (3.23)$$

and the pseudo-time step  $\Delta \tau$ , the final new shape of equation 3.3. choosing a 1st order scheme for the pseudo-time derivative discretization, is the following:

$$\left( \left( \frac{\Delta x_i}{\Delta t} + \frac{\Delta x_i}{\Delta \tau} \right) + \frac{\partial \vec{R}}{\partial \vec{Q}^*} (\vec{Q}^{*m}) \right) \delta \vec{Q}^* = -\vec{R}(\vec{Q}^{*m}) - \frac{\Delta x_i}{\Delta t} (\vec{Q}^{*m} - \vec{Q}^n) \quad (3.24)$$

in which  $\vec{Q}^{*m}$  is the unknown at the m-th subiteration of the pseudo-timestep whereas  $\vec{Q}^n$  is the known solution at the n-th iteration of the physical-timestep. Note that, since pseudo-time discretization can be of order 1 since accuracy is not required for the pseudo-timestep, for the physical timestep a higher order time-discretization scheme can be required in order to improve the quality of the solution. The basic approach starting at physical time step n ( $\vec{Q}^{m=0} = \vec{Q}^n$ ) is therefore to use an implicit scheme (Backward Euler) to integrate lhs of 3.24, until convergence is reached in m.

After iterating  $s$  times, the solution at time level  $n+1$  will be taken from most recent  $\vec{Q}^{m+1}$ . To be noticed is that the convergence of iterative process and the accuracy requirement are directly linked: if the iterative scheme fails to converge the error term will be indeed very large [21]. Even though a Dual-Time-Stepping scheme allows to run simulation with  $\Delta\tau$  arbitrarily high, a further refinement of the numerical CFL's evolution law was found to consistently improve convergence results. The code has been in fact structured in a way to work with the CFL input and never with the timestep input, this essentially in order to be able to bound the growth of CFL above the stability limits, both for explicit and implicit simulation. In order to reach pseudo steady-state the simulation might start from very abrupt transient, therefore an arbitrary choice of a fixed value for the  $\text{CFL}_N$ , that is the electrons CFL with  $\Delta\tau$  instead of  $\Delta t$ , might strongly compromise the convergence property of the method. A  $\text{CFL}_N$  evolution law called Residual Difference Method (RDM) has been used instead [9]:

$$\gamma_{k+1} = \begin{cases} \gamma_{\min} & \text{if } k < k_0 \\ \gamma_{\text{RDM}} \cdot \left( \frac{1}{|R_k - R_{k-1}|} \right)^{\alpha_{\text{RDM}}} & \text{if } k \geq k_0 \end{cases} \quad (3.25)$$

where  $k_0$  is the first index satisfying  $R_{k_0} \leq R_{k_0-1}$  and  $\gamma_{\min}$  is the starting  $\text{CFL}_N$ . The idea behind this formulation is the following: if the iterates seem to be converged, i.e., the solution does not change significantly between two consecutive iterations, one may increase the time step to check if the solution remains stable. On the other hand, if the solution varies much from one time step to another, smaller time steps should be chosen in order to resolve all flow features. Equation 3.25 comes from the fact that the difference of the solutions of two consecutive iterations is exactly the difference of their residuals: the smaller the difference of two consecutive residuals, the larger the  $\text{CFL}_N$  number for the next iteration. A fine tuning of parameters  $\gamma_{\text{RDM}}$  and  $\alpha_{\text{RDM}}$  will be presented in 5.3. A final parameter that will be introduced in this section in order to thoroughly define the pseudo-timestep sub-iteration is the relaxation factor  $\alpha_{\text{REL}}$ . Once found the solution at the  $m$ -th pseudo-timestep  $\vec{Q}^m$  the electrons solution is updated as follows:

$$\vec{Q}^{*m+1} = \vec{Q}^{*m} + \alpha_{\text{REL}} \delta \vec{Q}^{*m} \quad (3.26)$$

The reasoning behind equation 3.26 is the fact that, as already mentioned in section dedicated to implicit schemes, a linearized implicit iteration can be seen as a Newton's method iteration, with the only difference that within the Backward Euler scheme only one iteration was made, whereas in the DTS scheme sub-iterations are as many as to guarantee pseudo-time convergence. As shown in Fig. 3.6 it is of paramount importance to choose Newton's

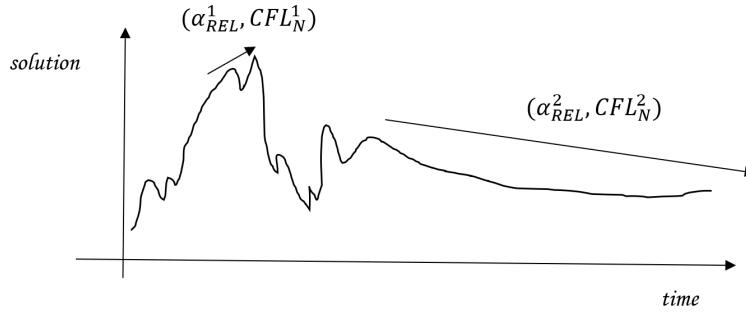


Figure 3.6: Different choice of Newton's parameter for strongly unsteady and weakly unsteady solution patterns

parameters  $\alpha_{REL}$  and  $CFL_N$  within the sub-iteration step in order to find a good trade-off between stability and convergence speed-up.

### 3.6.1.1 Local timestep

Now that thanks to the DTS technique the solution for electrons' dynamics at each physical timestep is a steady-state one, within Newton sub-iterations it is possible to exploit other numerical scheme used mostly in steady flows to further boost computational performance. Indeed one of these is method is the *Local timestep* scheme, for which the timestep is different in every cell and (in this case the pseudo-timestep  $\Delta\tau$ ) is updated according to the local  $CFL_N$ . For each cell in fact  $CFL_N$  is computed according to electrons mean velocity and sound speed in that precise cell, in this way different parts of the domain evolve with a different time scale than others: those in which flow is strongly unsteady will evolve faster w.r.t. other part of the domain in which flow is much closer to a steady state. This naturally implies that



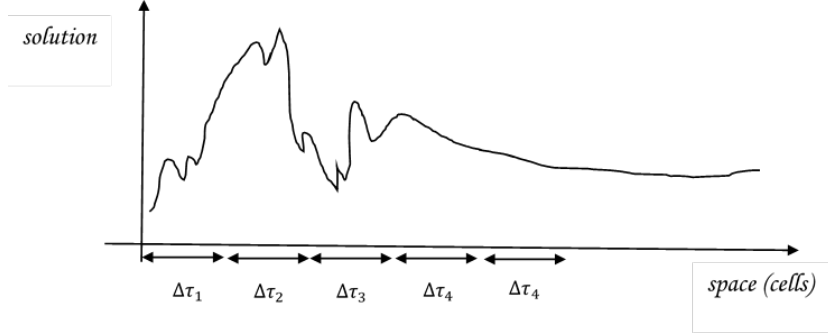


Figure 3.7: Local pseudo-timestep

pseudo-timestep  $\Delta\tau$  becomes local:

$$\left( \left( \frac{\Delta x_i}{\Delta t} + \frac{\Delta x_i}{\Delta \tau_i} \right) + \frac{\partial \vec{R}}{\partial \vec{Q}^*}(\vec{Q}^{*m}) \right) \delta \vec{Q}^* = -\vec{R}(\vec{Q}^{*m}) - \frac{\Delta x_i}{\Delta t} (\vec{Q}^{*m} - \vec{Q}^n) \quad (3.27)$$

### 3.6.1.2 Sub-iterations restart

One of the key additional features that has been introduced within this thesis project is the possibility to automatically restart a simulation whenever the choice of Newton's settings is not the most suitable anymore for that stage of the simulation and Newton's method is not converging anymore. The manifold of parameters to be changed could be arbitrarily large, in this thesis the choice was oriented to those parameters that, by trial and error, were found to work best:

- Starting  $CFL_N$

Very stiff initial conditions require a very small numerical CFL to start with, whereas implicit method is boosted the most when initial conditions are smoother

- $\alpha_{RDM}, \gamma_{RDM}$

Probably one of the most tricky and delicate parameters to be set: within this thesis optimal values are found to be both 1.1 and 1.1, but indeed

when a physical CFL can't be too high, in order to boost the speed of the implicit method those parameters can be chose in relation to the  $\alpha_{REL}$  as follows:

$$\begin{aligned}\alpha_{REL} \uparrow \alpha_{RDM}, \gamma_{RDM} \downarrow \\ \alpha_{REL} \downarrow \alpha_{RDM}, \gamma_{RDM} \uparrow\end{aligned}$$

- Convergence criteria

The main convergence criteria used inside the Newton's sub-iterations is the quadratic residual of electrons population:

$$\vec{R}_Q = \sqrt{\frac{\sum_i^{Cells} \vec{R}_i^2}{Cells}} \quad (3.28)$$

It has been noticed within this thesis project that restarting with a smaller residual tolerance improved the convergence property of the method.

- $\alpha_{REL}$

Choice of the relaxation factor depends essentially on the physical CFL: indeed the higher the CFL the smaller has to be the relaxation factor otherwise the risk is to take too long Newton's step and arrive at non-physical solution. Lower bound for this parameter is 0.3, whereas the upper one is 1.

- Reference variables

This was one of the most successful approach to improve the speed of the implicit method: indeed as mentioned in introductive section 2.4 it is not feasible to unitary nondimensionalize every single variable, hence some variables like for example electrons velocity along x-axis were found to be very sensitive to the reference variable they were scaled with. Scaling in fact with velocity two orders of magnitude higher than thermal velocity results into a more stable and definitely faster converging implicit method.

Instead of manually restarting the simulation with different parameters, the software *FORMICA* has been endowed with two possible alternative to being restarted once a NaN is found in the variables: a *static* one in which restarting parameters are read from input file, and a *dynamic* one in which parameters were updated scaling them up or down by a fraction of their starting values.

## Chapter 4

# Implementation and verification

In this section a more detailed overview of the software developed within the contest of this thesis project will be given, in particular all the features of the code will be presented and an *iteration summary* will be then presented. In the second part validation of the multi-fluid model and of the Poisson's equation solver will be presented through two different standard test-cases.

### 4.1 ***FORMICA*: Multi-fluid 1D Solver for collisional charged plasma**

The software is able to reproduce the following type of simulation:

- 1D multi- and single- fluid simulation for both charged species electrons and ions (velocity along y-axis is included as depending solely on x variable) (equations 2.38-2.43 and 3.16)
- Explicit (Midpoint Euler) or implicit (DTS) only 1st order in time
- Dimensional or nondimensional

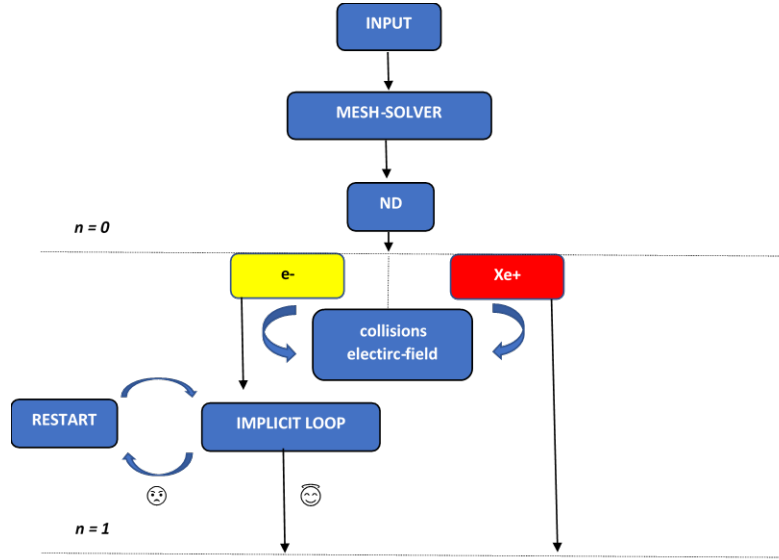


Figure 4.1: Physical iteration summary (DTS) for nondimensional (ND) simulation

- Fully collisional, with only elastic collisions only or no collisions at all
- Accounting for self-consistent electric field (electrons-ions interaction) or as a neutral gas flow
- With uniform or non-uniform (Chebyshev) mesh
- With restart option for both physical iterations (manual) and pseudo sub-iterations (automatic: static or dynamic)
- Possibility to run simulation with: isothermal closure for energy equation (electrons and ions)
- Possibility to run simulation with: profile (x) for magnetic field or constant
- Possibility to run simulation with: profile (x) for neutrals density or constant

A quick summary of an implicit (DTS) iteration is reported in Fig. 4.1.

test case	numerical	physics	EM	IC-BC
Mesh	$CFL_N$	$m_{e,i}, \gamma_{e,i}, q_{e,i}$	Voltage drop	$u_{e,i}, T_{e,i}, \rho_{e,i}$
Single-Multi	$CFL_{Phys,e,i}$	$\epsilon_{ex,iz}$	MAX $\mathbf{B}$	
ME-DTS	$\alpha_{REL}$	$n_n$	Poisson's BC	
Restart	$\alpha_{RDM}, \gamma_{RDM}$			
ND	residual tolerance			

Table 4.1: FORMICA input settings

## 4.2 Verification of multi-fluid model: 1D Sod-Shock

A very first verification for the sourceless multi-fluid test case is the Sod-Shock problem, in which a neutral gas<sup>1</sup> is such that its velocity is void everywhere but there is a density and pressure discontinuity at half domain, as summed up in Table 4.2. Domain is 1.1 m length and it is discretized

	LEFT	RIGHT
$\rho[kg/m^3]$	3.918	0.9795
$\tilde{u}[m/s]$	0	0
$[K]$	0.0066	0.0066

Table 4.2: Electrons BC Sod-Shock

	LEFT	RIGHT
$\rho[kg/m^3]$	4.008	2.002
$\tilde{u}[m/s]$	0	0
$[K]$	6203	6203

Table 4.3: Ions BC Sod-Shock

in 1000 cells. As far as initial conditions are concerned, boundary conditions from Tab. 4.3 are essentially prolonged until the half of the domain. The analytical solution can be found solving the Riemann problem at the interfaces [17], the studied flow presents enough critical features like discontinuities and

---

<sup>1</sup>electrons and ions not interacting though  $\mathbf{E}$  field can be considered as neutral gas

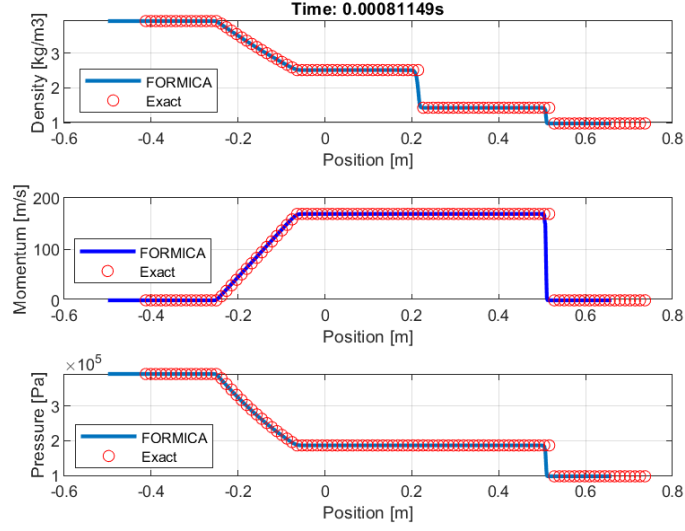


Figure 4.2: FORMICA electrons Sod-Shock

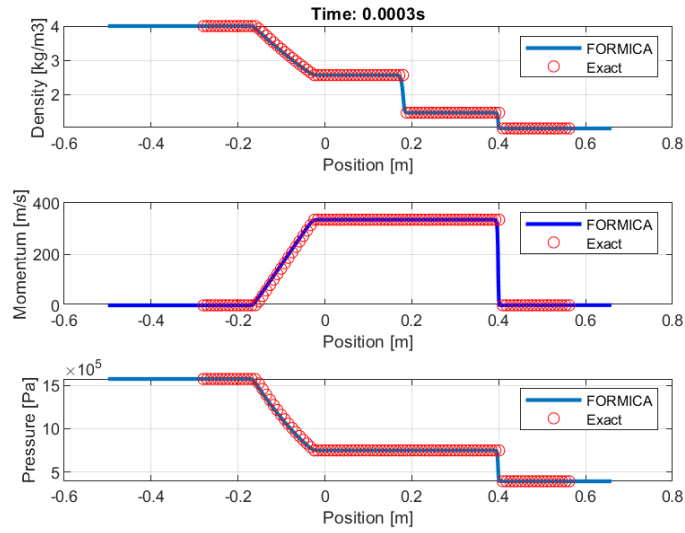


Figure 4.3: FORMICA ions Sod-Shock

rarefaction waves to be a good test to check the correctness of the fluxes implementation. Simulation was run explicitly using Midpoint Euler method, with CFL bound of 0.9. As it is possible to evince from Fig. 4.2 the software is able to reproduce faithfully the exact analytical solution. Analogous reasoning applies for ions population, whose simulation can run uncoupled to the one of electrons. Boundary and initial conditions are summarized in Table 4.3. Once again the software is able to reproduce the exact Riemann solution as shown in Fig. 4.3.

### 4.3 Verification of Poisson's solver: dispersion relations for longitudinal modes in plasma

When introducing the effect of the electric field  $\mathbf{E}$ , one way to check whether both the implemented algorithm and the code structure are correct, is to simply test the software to verify the so called *plasma dispersion relations*. A population of charged particle with precise density and temperature properties, when invested by an external electromagnetic wave, tends to oscillate according to a precise law that bounds spatial frequency  $k$  of the incoming wave with temporal frequency at which plasma starts to oscillates  $\omega$ . In order to know the shape of this law it is necessary to "ask" Maxwell set of equations which waves are admitted in the medium and which not: by combining the Fourier transformed Maxwell equations for electrodynamics one can define the following operator:

$$\Lambda = \left[ N^2 \left( \mathbf{I} - \frac{\vec{k} \otimes \vec{k}}{k^2} \right) - \epsilon \right] \quad (4.1)$$

where  $\epsilon$  is the dielectric tensor and  $N = c/\omega k$  is the refraction index, such that:

$$\Lambda \mathbf{E} = 0 \quad (4.2)$$

In order for equation 4.2 to have a non-trivial solution, the following condition must hold:  $\det \Lambda = 0$ . Solutions will show the so-called longitudinal and



transversal modes, that is which kind of law must bound  $k$  and  $\omega$  for the wave to propagate in the plasma. Here only longitudinal modes will be studied, and three cases will be taken in consideration: cold-plasma ( $T_e \approx 0K$ ), hot-plasma ( $T_e \approx 500000K$ ) and eventually cold plasma with constant external magnetic field. What is being done practically is to create a space dependent small perturbation in the initial number density variable, in order to associate to it a spatial frequency of the incoming electromagnetic wave:

$$n(x) = n_0 + n_0(\cos(2\pi x/L)) \times 10^{-5} \quad (4.3)$$

setting a total simulation time of about 100 ns, 1000 total cells ( enough to investigate reasonable wave numbers) and using periodic boundary conditions to simulate an infinite domain length. Electric field is then extracted

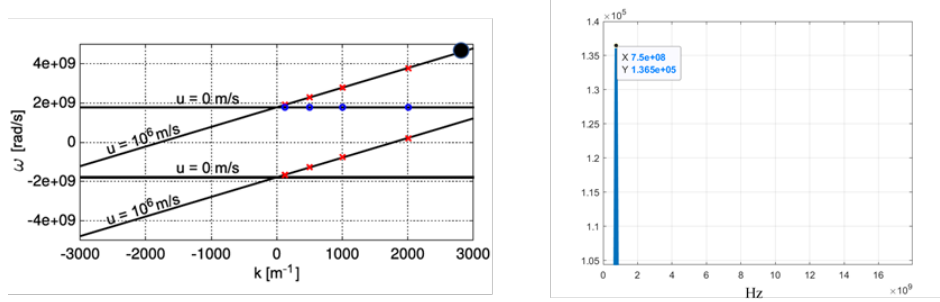


Figure 4.4: Cold-Plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT

at a certain location at each timestep and after each simulation a FFT is performed on it in order to extract the plasma frequencies (RHS of Figures 4.4, 4.5 and 4.6) and compare them with analytical results (black dots on the LHS of Figures 4.4, 4.5 and 4.6). The very first case is the one of cold plasma, whose dispersion relation coming out of 4.2 is the following:

$$\omega = k u_x \pm \omega_{pe} \quad (4.4)$$

where  $\omega_{pe}$  is the electrons plasma frequency. Results in picture 4.4 show that by setting a spatial frequency a bit less than 3000[1/m] and a background electrons velocity of the order of  $10^6[m/s]$ , the resulting plasma oscillations

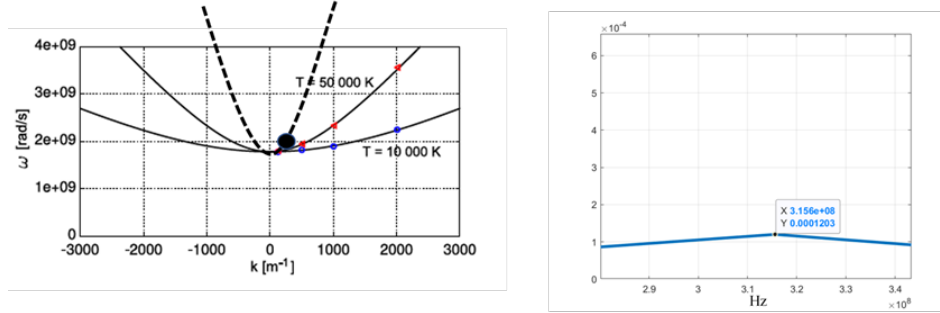


Figure 4.5: Hot-Plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT

are  $\omega_{pe} = f2\pi = 5[\text{rad/s}]$ . Increasing temperature of the plasma at about  $500000\text{K}$  for higher spatial frequency it is possible to observe wave with non-zero group velocity  $v_G = \frac{\partial\omega}{\partial k}$ , the dispersion relation reads then as follows:

$$\omega^2 = \omega_{pe}^2 + \frac{3}{2}k^2v_{th}^2 \quad (4.5)$$

where indeed the thermal velocity  $v_{th}$  coincides with the group velocity. In Fig.4.5 (where the dashed line corresponds to the isothermal of  $T = 500000\text{K}$ ) it is possible to see how also in this case, FORMICA is able to reproduce the behavior of hot plasma. The final test-case proposed within the Poisson's solver validation is the one of plasma subject to an electromagnetic wave with non-zero magnetic field. For this specific test case the magnetic field  $\mathbf{B}$  has been chosen external and constant equal to  $1\text{mT}$ . Plasma in the particular case is subject to a an-isotropic external force that will generate a coupled structure of the dielectric tensor: dispersion relation will read in such a way that electrons oscillate at *upper hybrid frequency*  $\omega_h$ . Results is once again shown in Fig. 4.6.

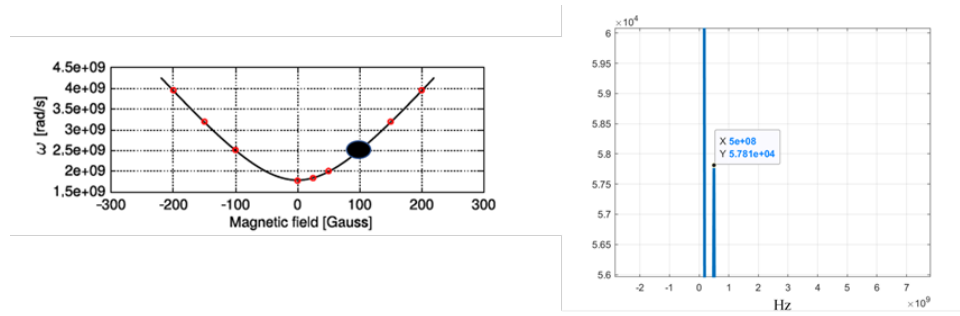


Figure 4.6:  $B$  into plasma: lhs analytical feedback versus FORMICA results (black circles); rhs FFT

# Chapter 5

## Results

In this final part of the thesis there will be discussed some of the obtained results. Attention will first be drawn to the reaction rate implementation discussed in a pure theoretical way in 2.3.1. Secondly the multi-fluid model will be tested on a well-known literature test-case [19]: collisionless plasma expanded into vacuum. The two conclusive test cases will have to do with a simplification of an Hall-Thruster engine and a full preliminary study of an Hall-Thruster engine endowed with radial magnetic field respectively. The former case will also be used to test the implementation of the Dual-Time-Stepping scheme.

### 5.1 Numerical estimates of reaction rates

For each collisional term introduced in section 2.3.1, a numerical computation of a reaction rate through the integral in equation 2.36 is carried out using a standard trapezoidal rule for numerical integration: one integral for each temperature within the physical range of a Hall-Thruster engine ( the chosen range in this thesis project is from 50 K up to  $1,2 \times 10^6$  K, according to literature [1]. Values of different scattering cross-sections  $\sigma$  as a function of energy levels were eventually taken from reference [18]. While processing the integrals, one thing not to be neglect is the choice of a proper energy

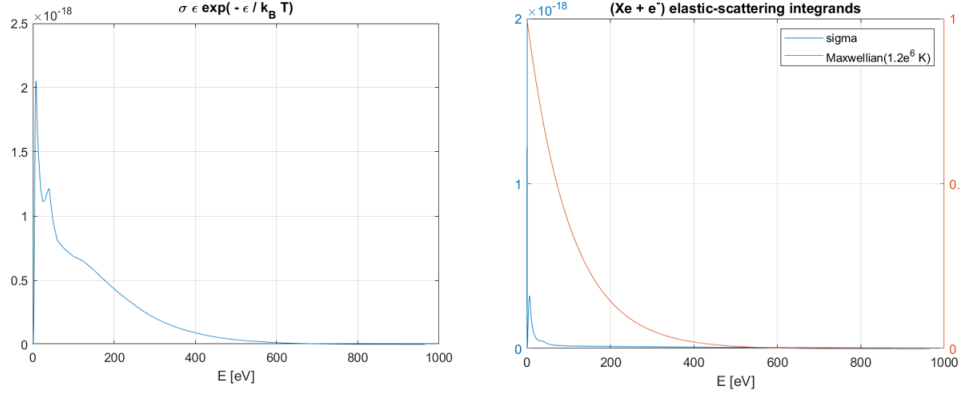


Figure 5.1: LHS: Energy range check for integrand RHS:  $\sigma$  and Maxwellian plot

range for numerical integration: indeed the integral bounds are 0 and  $\infty$  for equation 2.36, but being for obvious reasons not possible to have such a precise range bound, a good approximation can be to take the highest energy value there where the tail of the integrand is essentially negligible, see LHS of Fig.5.1. This procedure has to be done for every temperature. A further

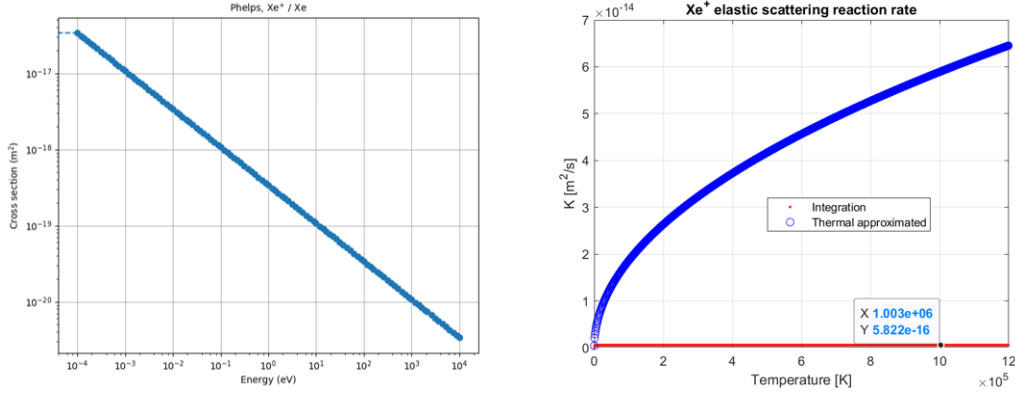


Figure 5.2: Ions - neutrals elastic scattering

check can be also done comparing the shape of the scattering cross section and the one of the Maxwellian distribution for a given temperature, RHS of Fig.5.1. In Figures 5.2,5.3,5.4,5.5 it is shown on the left hand side the

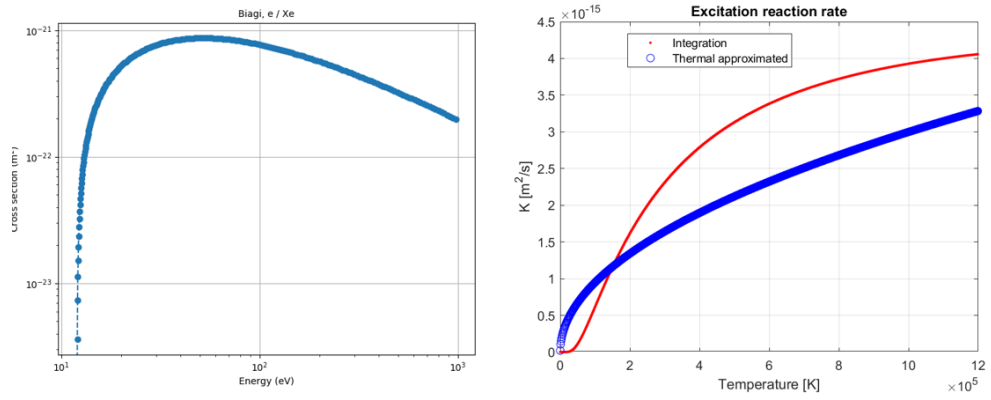


Figure 5.3: Excitation

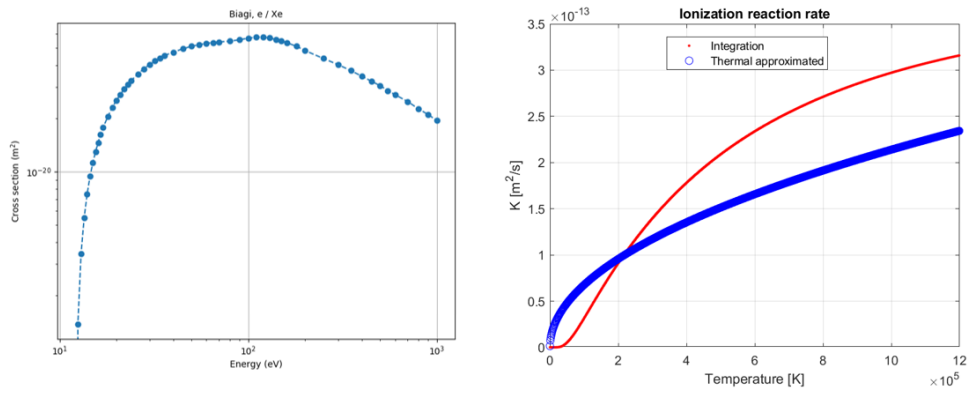


Figure 5.4: Ionization

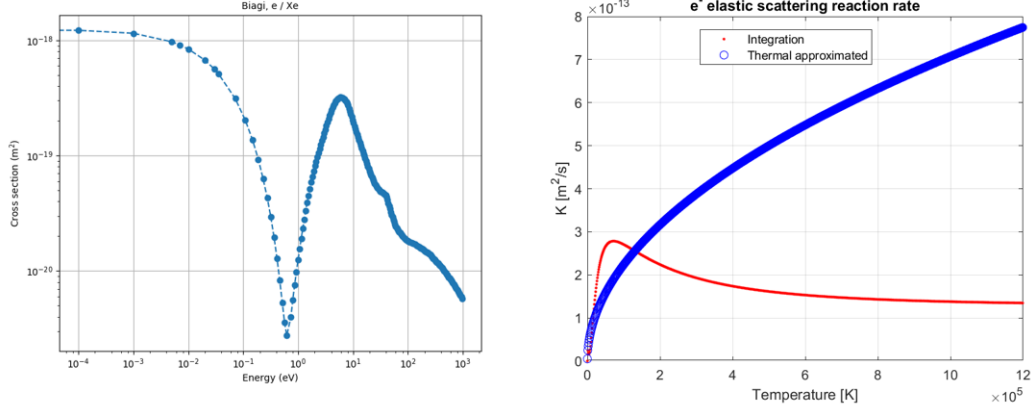


Figure 5.5: Electrons - neutrals elastic scattering

values of the scattering cross-section from [18], whereas on the right hand side a comparison between the analytical derivation proposed by [16] of the reaction rates, and the thermal approximation introduced at the beginning of the section using only a thermal velocity and an averaged scattering cross-section to retrieve an approximated reaction rate: it is possible indeed to see how in certain cases like ionization and excitation the approximation still is not that wrong, but in some other cases like elastic scattering in which it returns very incorrect estimates. Reaction rates are then inserted into collisional frequency terms for each different type of scattering according to equations seen in section 2.3.1.

## 5.2 Plasma expansion into vacuum

Studying dynamics of plasma left expanding into vacuum is not just a useful test case to try a totally new approach with respect to those already proposed in literature (mainly to check if a multi-fluid solver is actually able to reproduce with precision same results), but it also allows to introduce a preliminary study of behavior of Hall-Thrusters that, as a matter of fact, expand plasma into vacuum during their operational life in space. The analytical derivation of the physical model that this thesis project will follow is

in reference [19]. At initial time plasma is assumed to occupy only half-space: ions are cold<sup>1</sup> and initially at rest and in a quasi-neutral state with electrons. Electrons density is defined through a Boltzmann distribution as follows:

$$n_e = n_{e0} e^{\frac{q_e \phi}{k_B T_e}} \quad (5.1)$$

where  $\phi$  is the usual electrostatic potential, void at  $-\infty$  and such that the already introduced Poisson's equation holds. For  $t > 0$  electrons are assumed to stay in equilibrium with potential  $\phi$ , such that both equation 5.1 and Poisson's equation still hold, whereas to reproduce ions expansion continuity and momentum equations are presented in the following fashion:

$$\left( \frac{\partial}{\partial t} + \frac{v_i \partial}{\partial x} \right) n_i = -n_i \frac{\partial v_i}{\partial x} \quad (5.2)$$

$$\left( \frac{\partial}{\partial t} + \frac{v_i \partial}{\partial x} \right) v_i = - \left( \frac{Z q_i}{m_i} \right) \frac{\partial \Phi}{\partial x} \quad (5.3)$$

By integrating Poisson's equation from ion-front to  $+\infty$  one can verify that for  $t = 0$  and for  $\omega_{pi} t \gg 1$  a very precise formula for the electric field at ion front is the following:

$$E_{front} \approx 2 \frac{E_0}{\sqrt{2e_N + \omega_{pi}^2 t^2}} \quad (5.4)$$

From this result it is possible to obtain , after further integration procedure, the analytical formula of the ion-front velocity:

$$v_{front} \approx c_s \ln(\tau + \sqrt{\tau^2 + 1}) \quad (5.5)$$

where  $c_s$  is the sound speed of ions, and  $\tau = \omega_{pi} t / \sqrt{2e_N}$ . The main two analytical feedback that will be use to test the software are therefore those of equations 5.4 and 5.5. Before presenting a comparison of numerical results

---

<sup>1</sup>Ions pressure into momentum equation is therefore not considered



with the analytical solution, it is worth pointing out that the FORMICA multi-fluid solver is solving a different set of equations with respect to those presented for this test-cases, not only as far as ions are concerned but especially for electrons population. For the current simulation the initial and boundary conditions are summed-up in Table 5.1 , where the right and left boundary extends respectively until the half of the physical domain. In Ta-

	LEFT	RIGHT
$\rho_e[kg/m^3]$	$1.82187 \cdot 10^{-13}$	$1.82187 \cdot 10^{-16}$
$u_e[m/s]$	0	0
$T_e[K]$	300	300
$\rho_i[kg/m^3]$	$8.2839 \cdot 10^{-4}$	$8.2839 \cdot 10^{-7}$
$u_i[m/s]$	0	0
$T_i[K]$	0.03	0.03

Table 5.1: Plasma expansion into vacuum: BC

ble 5.1 is to be noticed that: vacuum is enforced by scaling RHS values by a factor 1000, and that ions temperature is way below the Fermi temperature limit for not consider quantum mechanics adjustment. Nevertheless the main purpose of this test case is a further validation of the multi-fluid model under a mere mathematical point-of-view. No collisions have been

$\lambda_D$	$\omega_{pe}$	$dt$
$8.45 \cdot 10^{-5}[m]$	$7.9782 \cdot 10^8[rad/s]$	$7.8754 \cdot 10^{-9}$

Table 5.2: Plasma expansion into vacuum: numerical settings

taken into account and the total time of the simulation was around  $10 \mu s$ , long enough for being in the analytical range mentioned few lines above and to appreciate a distinct ions-expansion front, but short enough to be able to appreciate some good results with an explicit scheme ( $CFL_{el} \approx 0.7$ ) without having to run too long simulations. Domain length is 0.036 m, whereas number of cells for this particular simulation was 9000, even though for sensitivity analysis even a higher number was employed. The reason behind this choice stands in one of the argumentation brought about already in section 2.2, when the concept of Debye length was firstly introduced (equation 2.5): indeed if charge separation wants to be studied, and hence a non zero source term for the Poisson's equation, the numerical scheme must be endowed with

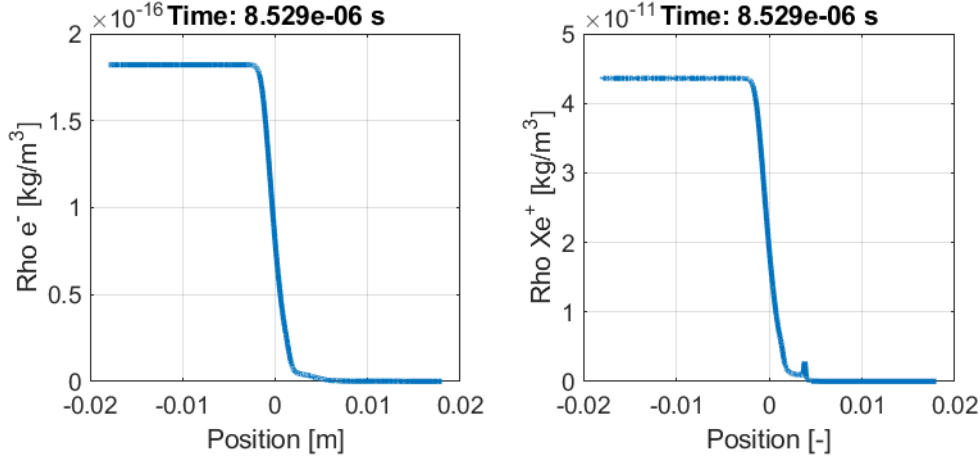


Figure 5.6: Electrons and ions density evolution

the possibility to resolve to physics to such a small scale length. In other words the size of the mesh can't be arbitrary, but it must be such that the domain's cell  $\Delta x$  has to be smaller than the Debye length  $\lambda_D$ , consequently the domain size has to be orders of magnitude ( at least 100 ) larger than the Debye length. At the same time, once imposed the physical CFL and the cell size (by forcing the just mentioned Debye length constraint), the timestep has anyway to abide by the physical time one is interested in studying: if the numerical simulation has been sampled with a timestep much greater than the characteristic time for electrons dynamics (  $\frac{1}{f_{el}} = \frac{2\pi}{\omega_{pe}}$  ), then no proper electrons plasma frequency will be caught by the numerical scheme. A quick summary of the numerical parameters to be set within this test case is presented in Table 5.2. From Fig.5.6 it is possible to see how both density profiles of ions and electrons evolve in time a after bit less than  $10\mu s$ : ions profile shows a bit after the first half of the domain already a small pick, denoting the expanding wave into the left hand side of the domain. In Fig. 5.7 a plot showing evolution of electric field (LHS): indeed electrons expand almost immediately towards the empty part of the domain, creating thus a charge imbalance that attracts ions through the self-consistent electric field. Ions therefore feel a second acceleration, other than the one due to the pressure gradient caused by the vacuum on the right part of the domain, due to an arising electric field that peaks there where the ion front is located. On

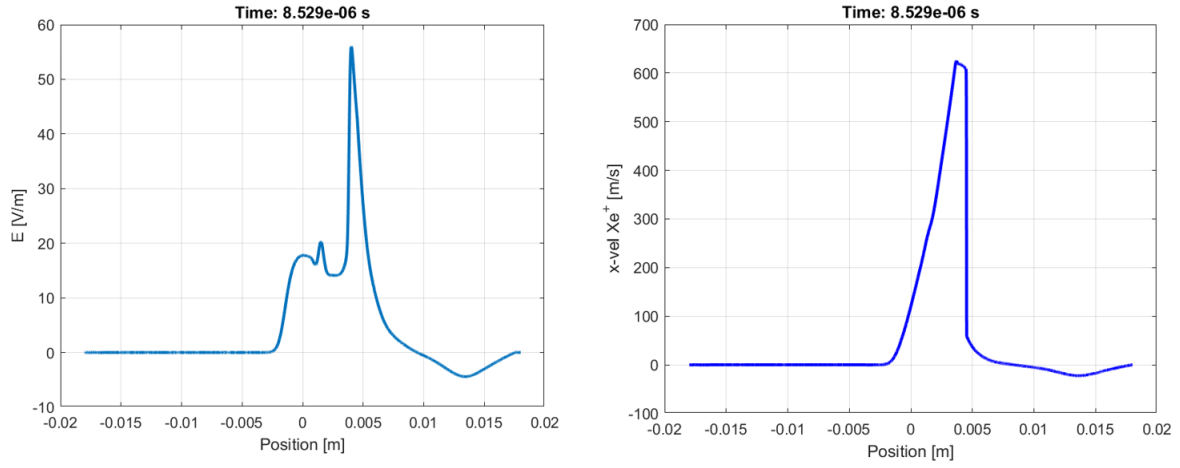


Figure 5.7: Electric field and ions axial speed

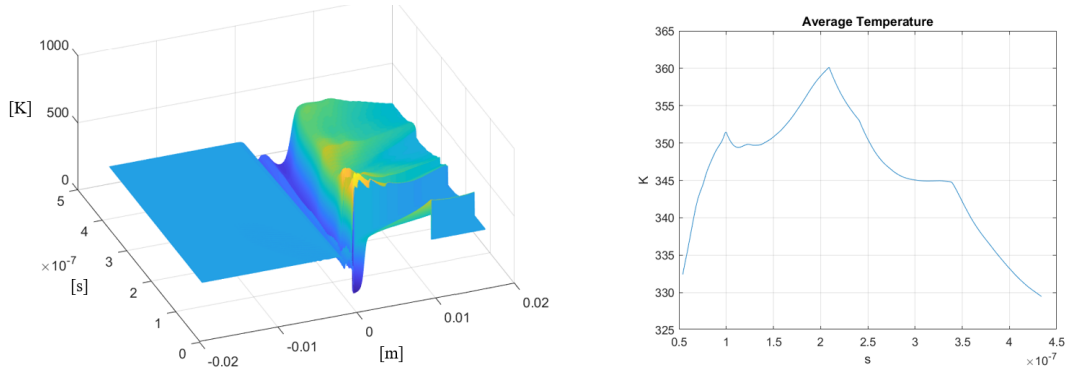


Figure 5.8: LHS Electrons temperature profile in space and time RHS Average temperature in time at ions expansion front

the RHS of Fig. 5.7 it is hence shown ions' peak velocity at the expansion front. To be noticed that ions' acceleration is due almost entirely by the

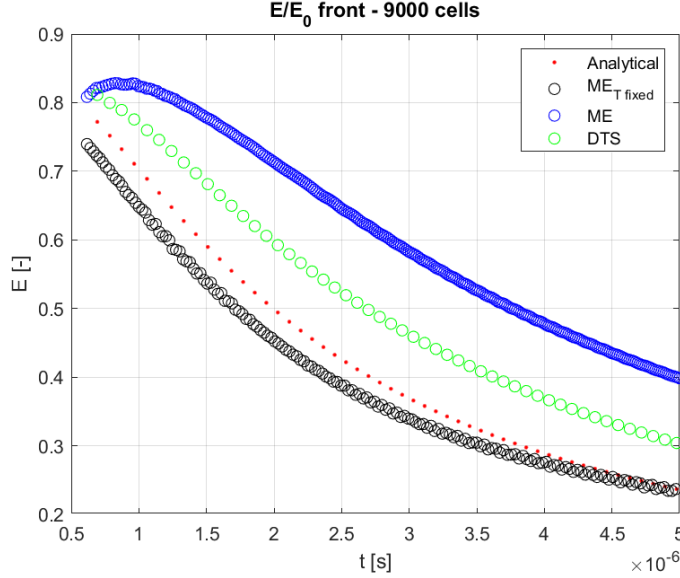


Figure 5.9: Scaled electric field front: results for Midpoint Euler (also with  $T_{el}$  imposed) and implicit scheme (DTS)

only presence of the self-consistent electric field: indeed no voltage drop is applied at both sides. Under the numerical point of view the challenge was to force a "no-current" flow condition on both sides: homogeneous Neumann conditions on both sides resulted in strong numerical instabilities due to the fact that potential was left pointwise undefined, since depends on charge separation through Laplace operator. Therefore homogeneous Neumann was enforced only on the RHS, leaving homogeneous Dirichlet on the LHS. As it is possible to infer from LHS of Fig. 5.7 no current is flowing on both extremes of the domain. Neither momentum nor energy equations are solved for electrons within the analytical solution presented, where electrons temperature is kept constant equal to  $300K$ . By running a multi-fluid simulation with FORMICA it is possible to observe that Temperature is not constant neither in space nor in time (Fig. 5.8). Once acknowledged that in the following results it will be presented also a case in which an isothermal closure for electrons energy will be enforced, instead of solving adiabaticity for the energy equation of electrons. In Fig. 5.9 a comparison among analytical

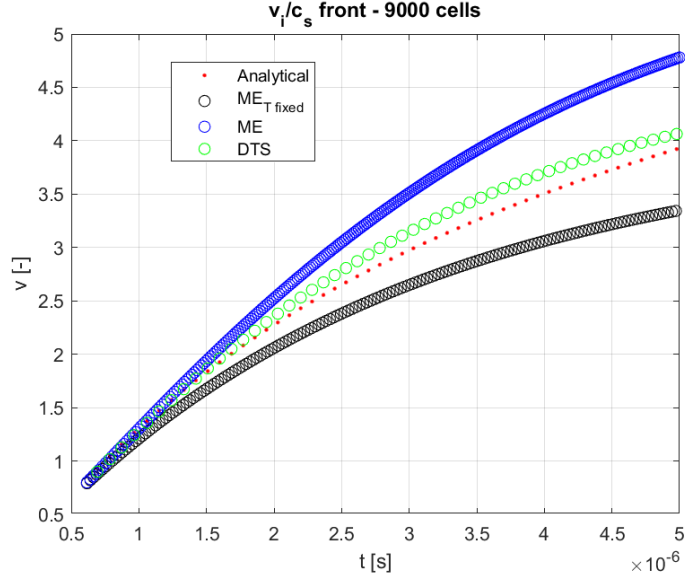


Figure 5.10: Scaled ions velocity front: results for Midpoint Euler (also with  $T_{el}$  imposed) and implicit scheme (DTS)

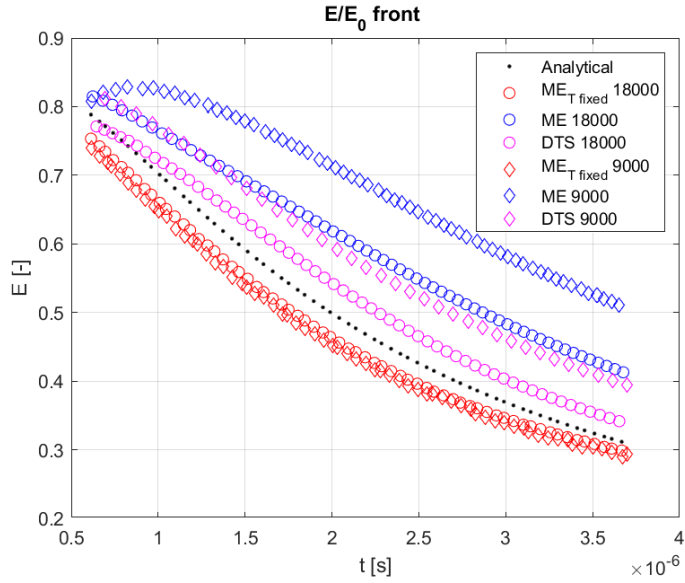


Figure 5.11: Sensitivity analysis electric field front

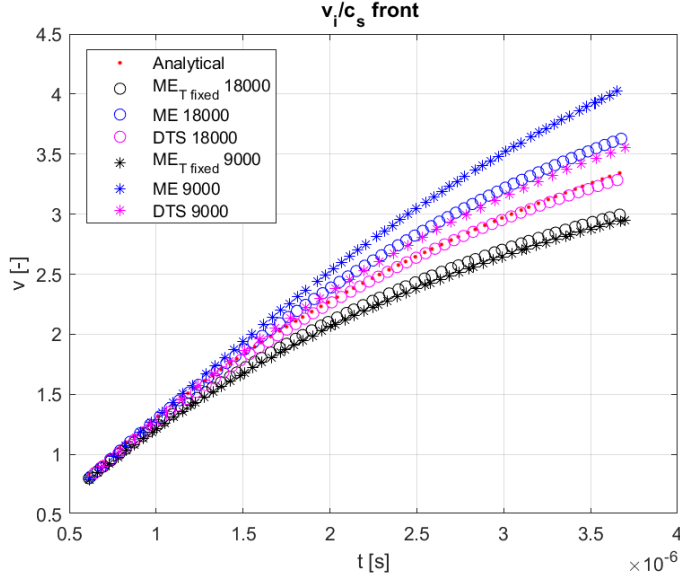


Figure 5.12: Sensitivity analysis ions velocity front

solution (eq. 5.4), numerical solution explicit with Midpoint Euler, explicit with Midpoint Euler and isothermal closure and implicit DTS solution. The one getting closer to the analytical solution is the numerical solution in which an isothermal closure were imposed, whereas it is also possible to infer the more dissipative behavior of an implicit scheme (whose technical details will be investigated in next section) with respect to an explicit one: indeed more dissipation mean a higher friction that reduce electric-field induced accelerations. Analogous result can be obtained for the ions' velocity front profile, for which isothermal and implicit schemes still outperform the explicit adiabatic one. In the last two figures, 5.11 and 5.12, a sensitivity analysis is proposed in which numerical results with refined mesh are plotted against the analytical solution.

### 5.3 Hall-Thruster engine without magnetic field: performance analysis on DTS method

Within this section it will be presented a simplified 1D model for a STP-100 Hall-Thruster engine [8] in which no radial magnetic field  $\mathbf{B}$  will be included. Indeed this simplification is introduced since the main goal of this analysis is to have a solid testcase in order to be able to evaluate performances of the DTS scheme versus those of an ordinary explicit scheme (Midpoint Euler). For what concerns the geometry the engine has an inner radius  $R_1 = 3.5$  cm

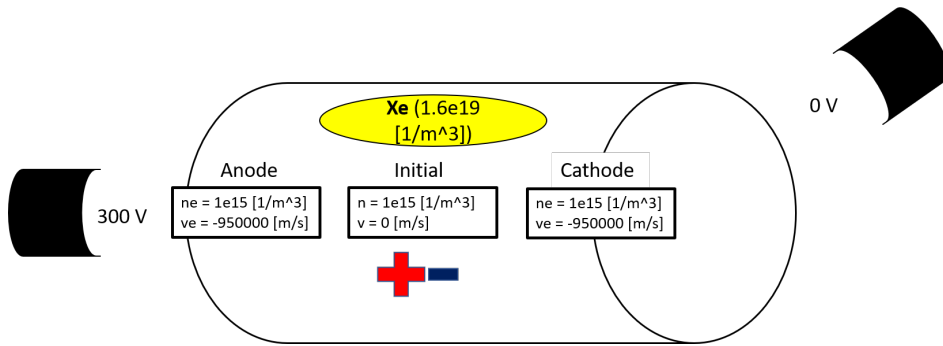


Figure 5.13: Hall-Thruster with no  $\mathbf{B}$ : boundary and initial conditions

and external radius of  $R_2 = 5$  cm, and with a channel width  $H = 1.5$  cm it is possible to define the cross sectional area  $A_c = \pi H(R_2 + R_1) \approx 40$  cm<sup>2</sup>. From anode to cathode voltage drop is around 300 Volt and a total discharge current of 4.5 A. Now given a Xenon gas flow rate from anode  $\dot{m} = 5.3$  mg/s, the ejected ion current is estimated of being  $I_i \approx 3.9$  A, meaning that that of electrons at cathode is around  $I_{el} = 0.6$  A. This last datum is important in order to define boundary conditions at the cathode since:

$$\vec{I}_{el,ca} = n_{el,ca} \vec{u}_{el,ca} A_c = 0.6A \quad (5.6)$$

meaning that if electrons density at the cathode is of the order of  $n_{el,ca} = 10^{15}$  [1/m<sup>3</sup>], then incoming velocity of electrons is around  $\vec{u}_{el,ca} = -9.8 \times 10^5$  [m/s]. By current continuity the same velocity is enforced at the anode as far as electrons are concerned. Warm electrons (50000 K as boundary and initial condition) therefore enter the cathode and ionize Xenon gas atoms,

$CFL_{Physical,el}$	100
$\alpha_{REL}$	0.3
$CFL_{Num,el}$	20
Residual tolerance	-5.0
$\alpha_{RDM}$	1.1
$\gamma_{RDM}$	1.1

Table 5.3: DTS parameters setting for no **B** Hall-Thruster testcase

$CFL_{Physical,el}$	10	20	100	100
$\alpha_{REL}$	0.5	0.4	0.4	0.3
$CFL_{Num,el}$	100	100	50	20
Residual tolerance	-5.0	-5.0	-5.0	-5.0
$\alpha_{RDM}$	1.3	1.2	0.7	1.1
$\gamma_{RDM}$	1.3	1.2	0.7	1.1
Time [minutes]	45	30	19	15

Table 5.4: Different numerical settings for DTS sub-iterations

whose density is kept constant at the value  $n_{Xe} = 1.6 \times 10^{19}$ . Inside the engine ions and electrons are still, in a quasi-neutrality state and initialized at a density of about  $n_{el} = n_i = 10^{15}$ , this in order not to deal immediately with strong density gradients at the extremes. Ions within this simulations are treated in such a way that only initial conditions are enforced, whereas boundaries should interfere the least possible with ions dynamics: for this reason homogeneous Neumann boundary conditions were enforced at both sides. A quick summary of boundaries and initial conditions is shown in Fig. 5.13. Domain length will be in this test case the distance between anode and cathode of a STP-100 Hall-Thruster engine [8], whereas the number of cells used is 725 (in order to respect the Debye length constraint mentioned in section 5.2). The explicit simulation was run with a maximum CFL of 0.9. When structuring the parameters for the DTS simulations things are a bit more delicate: it is worth to mention that for the following choice of parameters many simulations with many different combinations of settings were tried until a suitable one was chosen and results to be the best performing<sup>2</sup>. As it is shown in Figures 5.14, 5.15 and 5.16 FORMICA is able

---

<sup>2</sup>indeed among those that were tried out



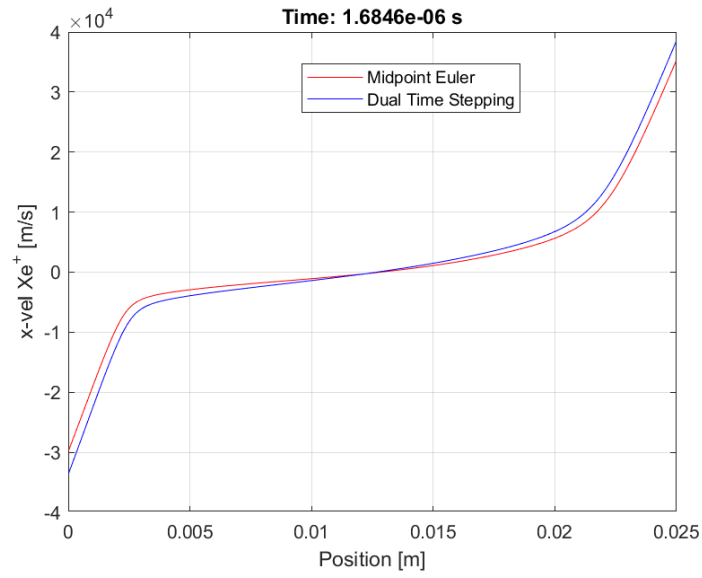


Figure 5.14: Ions axial velocity: ME vs DTS

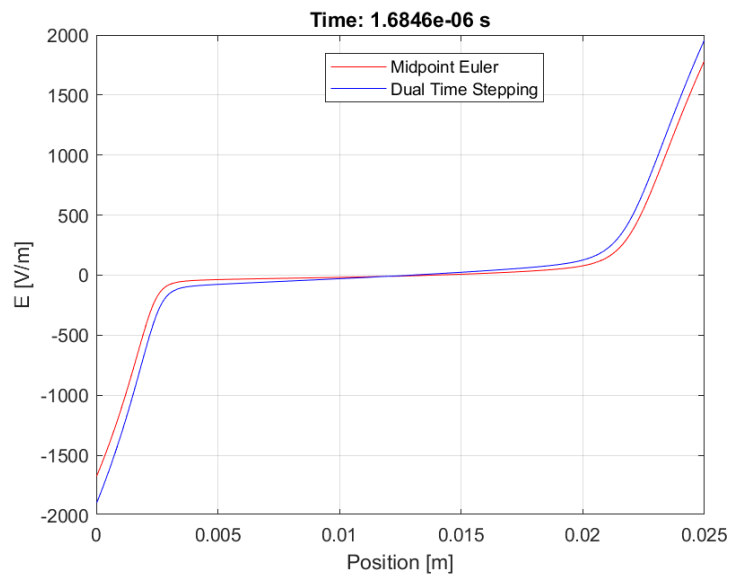


Figure 5.15: Electric Field: ME vs DTS

through DTS scheme to reproduce almost perfectly the result proposed by the Midpoint Euler scheme: a partial discrepancy in the solution might be due both to a more diffusive behavior of the implicit scheme (as already noticed in the previous test case (section 5.2), but especially because going with different timesteps, it was hard to sample exactly the same time frame. A quick analysis shows that in a specific case of a Hall-Thruster without  $\mathbf{B}$ , ions are accelerated mainly at the boundaries, where a sheath arises due to a charge imbalance (side peaks of  $\mathbf{E}$  in Fig.5.15). No magnetic field means no

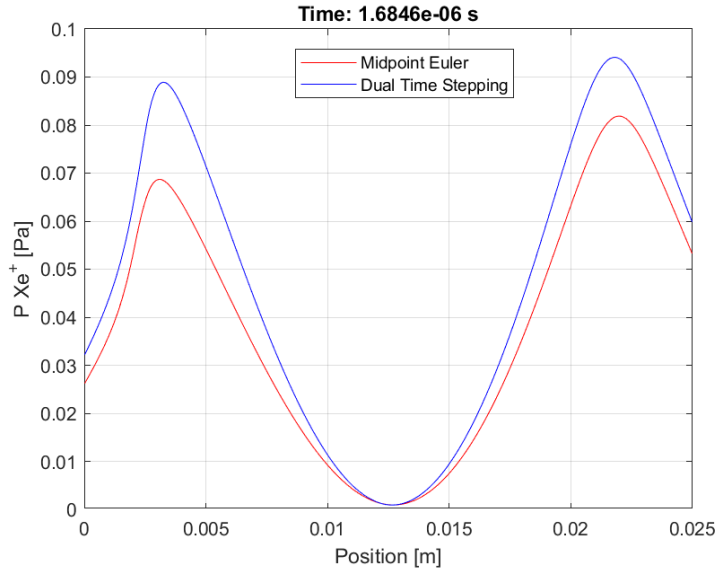


Figure 5.16: Ions Pressure: ME vs DTS

reduction in electron conductivity and hence no self-consistent electric field that accelerates ions at the engine's exhaust. Both (explicit and implicit) simulations were run for a physical time of about  $1.6\mu s$ , and as recapped in Tab. 5.3 the implicit one was able to run with a physical electrons CFL more than 100 times higher than the one set for the explicit one. Due to its very unsteady features a starting  $CFL_N$  not higher than 20 was chosen: for the very first part the amount of sub-iterations averaged in fact 50, but after a first transient they dropped down to 6 in the final part of the simulation where a steady-state starts to be approached. Under a computational point of view Midpoint Euler simulation took around 70 minutes to reach  $1.6\mu s$ , whereas DTS took only a bit more than 15 minutes, with a time saving of

about 5 times. Some of different parameters combinations tried and their results are illustrated in 5.4.

## 5.4 Preliminary results for Hall-Thruster engine with radial magnetic field $B$

The very last testcase presented within this thesis framework is a Hall-Thruster engine STP-100 fully endowed with a radial magnetic field  $B$  [8]. It is important to underline that section is not presented as a completely

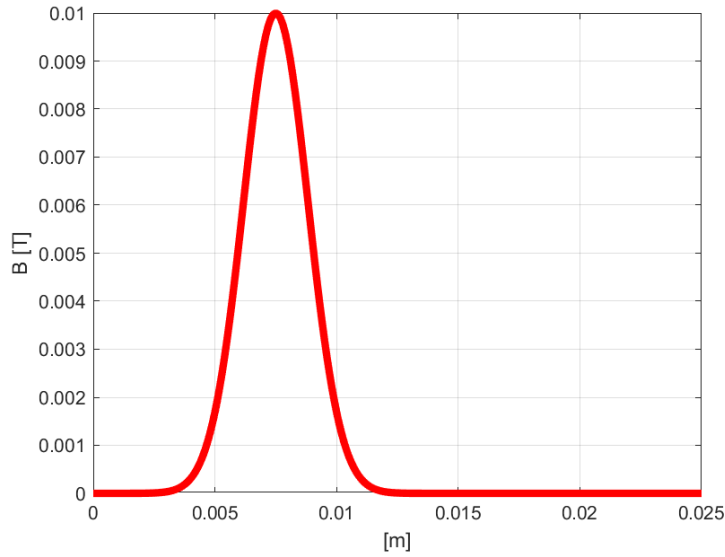


Figure 5.17:  $B$  profile

achieved result, but rather as a preliminary one since studying plasma devices with the aim of properly simulating a throughout electrons dynamic with a fluid-model approach is all but a trivial task. Moreover a neutrals dynamic was not provided by the multi-fluid solver model, meaning that it was not possible to retrieve an essential information when studying ions dynamics, that is the evolution of the ionization front. Having put this straight what is going to be presented in this last section are just few of the sev-

eral tried configurations aimed at simulating a correct dynamics within a Hall-Thruster engine. Domain length is 0.25 m, similarly to the test case

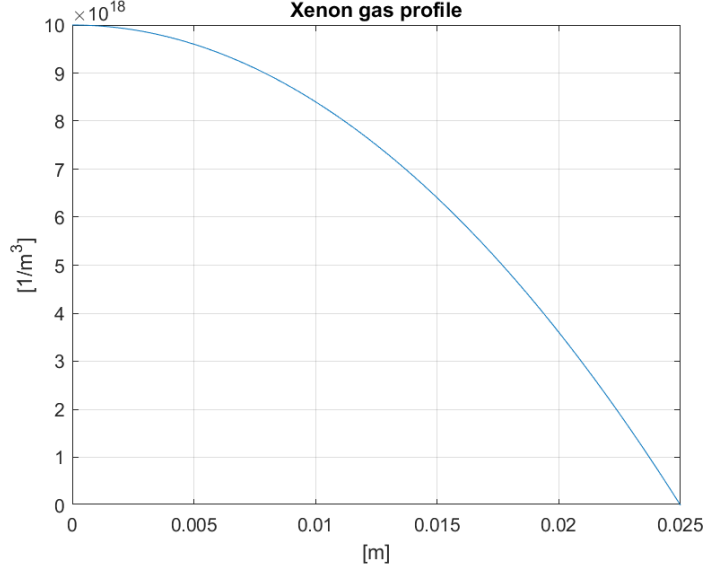


Figure 5.18: Xenon gas steady-state profile

without external magnetic field, whereas the number of cells abiding by the Debye length criteria is 1000. The very first thing to set is the shape of the magnetic field (Fig. 5.17 ): its peak is situated at the engine exhaust, at about  $7.5mm$  from the anode [6], whereas the range of maximum magnetic field values goes from  $0.005T$  up to  $0.02T$  [7]. One of the reason behind this choice for the  $\mathbf{B}$  shape is to maximize the efficiency extraction of ions from the device, and this happens if the acceleration region is thin and if it is collocated not too far from exhaust. A second feature added to the solver is

	Anode	Initial	Cathode
$n [1/m^3]$	$10^{12} : 10^{12}$	$10^{15} : 10^{15}$	$1^{15} : 10^{12}$
$T [K]$	$25000 : 5000$	$25000 : 5000$	$25000 : 5000$
$u_x [m/s]$	$-9.8 \times 10^5 : 0$	$0 : 0$	$-9.8 \times 10^5 : 0$
Voltage	200		0

Table 5.5: Initial and boundary conditions for  $\mathbf{B}$  Hall-Thruster *Case 1* (electrons:ions)

the possibility to simulate a likely shape of neutrals profile at steady-state, in order not to keep the same non physical constant profile all over the engine liner, a possible neutrals profile is introduced in Fig. 5.18 [7]. For

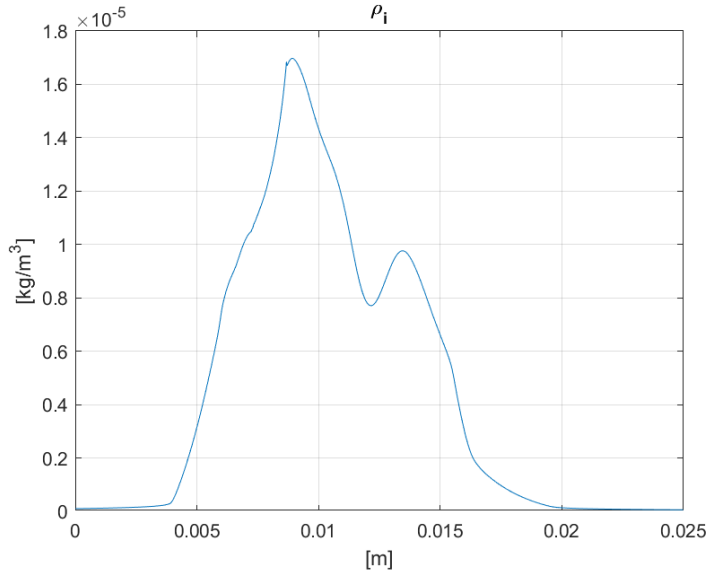


Figure 5.19:  $\rho_i$  in *Case 1*

what concerns initial conditions the strategy of starting with higher uniform electrons and ions density ( $10^{17}$  [ $1/\text{m}^3$ ]) than the "no- $\mathbf{B}$ " case was tried out in order to get as close as possible to a steady state value of  $10^{19}$  [ $1/\text{m}^3$ ] registered experimentally close to the exhaust region ([1],[7],[8],[6]): increasing number density though leaving temperature unchanged means tightening the constraint on the Debye length, and pushing the need to simulate with a much more refined grid. Results were that a finer grid is able to capture even small density gradient, that were unavoidably amplified by the Poisson's solver resulting in very unsteady and oscillating simulations. Boundary conditions were modified in order to create a sort of vacuum condition for electrons at the anode (therefore introducing density gradients that in this case one could not avoid not solve), whereas ions were not solved anymore with homogeneous Neumann and set with very low density both at cathode and anode. The reason behind this choice was that thanks to the property of Euler equations, each cells eigenvalues study is able to decide whether, according to the flow regime, information is coming in or going out of the

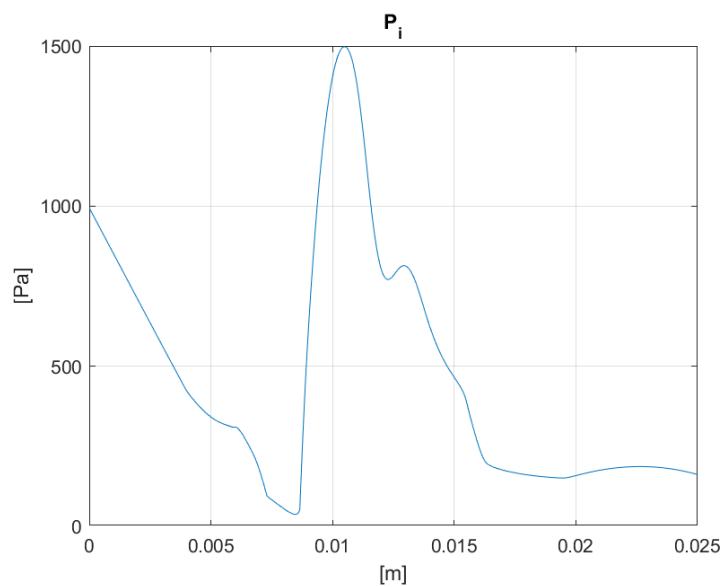


Figure 5.20:  $P_i$  in *Case 1*

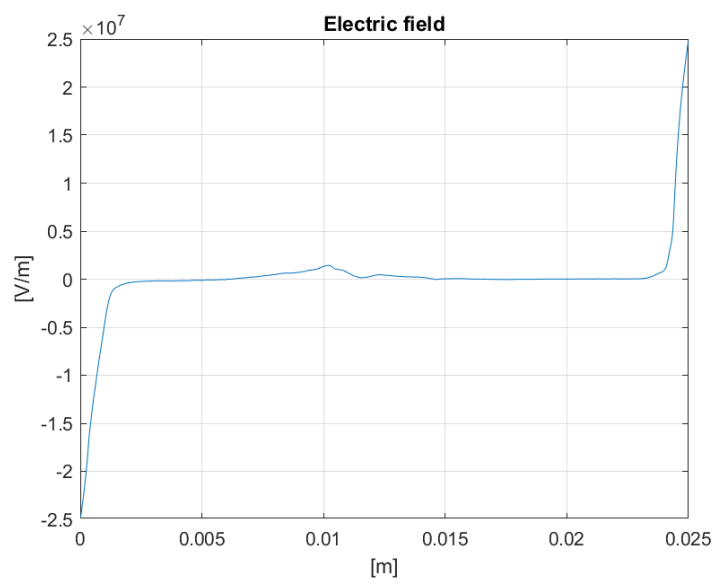


Figure 5.21:  $\mathbf{E}$  in *Case 2*

boundary ghost cell. One of the main issues connected with the introduction of magnetic field was the not-proportional increase in the y-momentum with respect to the one of total energy density: indeed magnetic field effect is such that particles start to drift along the azimuthal direction (in the local frame of reference y-axis), increasing hence momentum of ions along this direction. Magnetic field does not contribute to system energy though, and without an additional source of energy for ions, pressure drops to nonphysical values, generating thus oscillations and instabilities near the exhaust region. Few

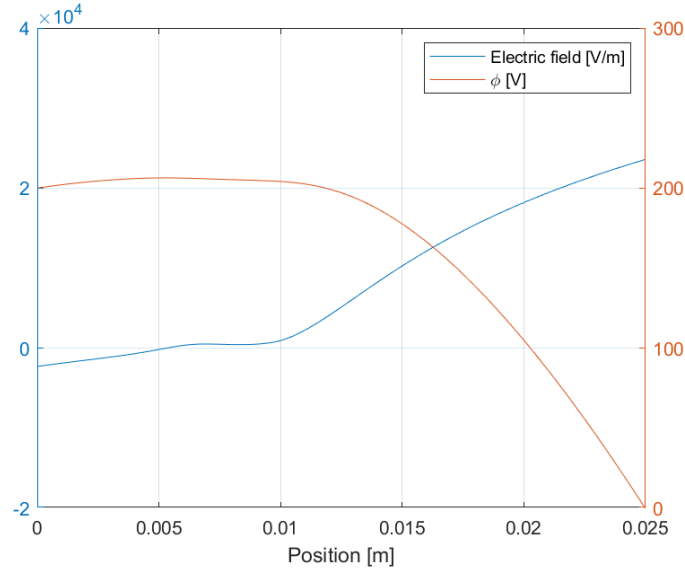


Figure 5.22:  $\phi$  and  $\mathbf{E}$  in *Case 3*

alternative were proposed to curb this issue within this thesis framework: the first one was to increase the elastic collisional frequency of ions (about 100 times), in order to absorb part of the magnetic field impact (*Case 1*), a second one was to push ions and electrons temperature much higher at the initial state and at boundaries (*Case 2*). A third and final one proposed a total different energy closure in which ions temperature was kept fixed (*Case 3*). The results about to be presented are the ones whose solution profile was the closest (but still far from being the exact one [6]) obtained with a full multi-solver simulation at steady state in *Case 1*: initial and boundary conditions are summed up in Table 5.5, whereas  $\mathbf{B}$  and  $n_{Xe}(x)$  profiles are

shown in equations 5.7 and 5.8, Figures 5.17 and 5.18 respectively.

$$\mathbf{B}(x) = 100e^{-16(\frac{x}{0.025}-1)^2} \quad (5.7)$$

$$n(x) = -1.59 \times 10^{22}x^2 + 10^{19} \quad (5.8)$$

Pressure and density of ions are shown in Figures 5.19 and 5.20, showing a pick close to the exhaust region and lower values in the tails close to the domain extremes. In *Case 2*, that differs from *Case 1* only for initial and boundary temperatures of electrons and ions, set respectively to 100000K and 10000K, and for the fact that elastic collisional frequency of ions has not been augmented. Fig. 5.21 shows how electric field profile at steady state

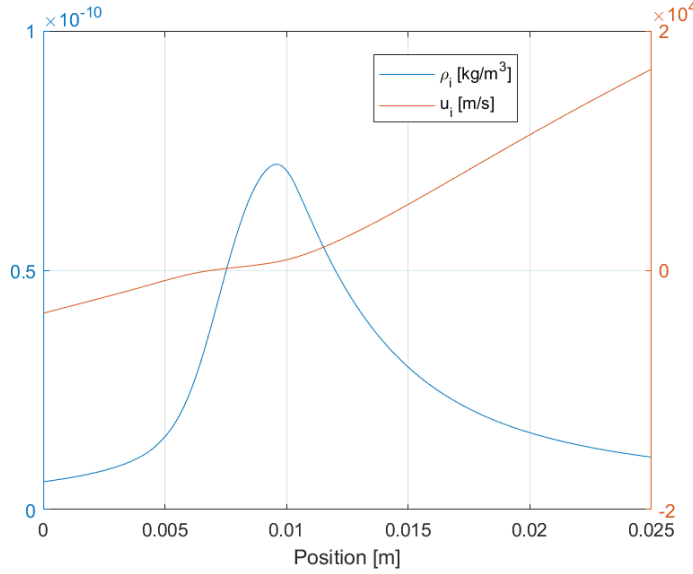


Figure 5.23:  $\rho_i$  and  $u_i$  in *Case 3*

presented not only the two sided sheaths already spotted in the "no- $\mathbf{B}$ " case, but also a second electric field pick right close to the exhaust region, there where ions are accelerated. Steady state results of *Case 3* are eventually shown in Figures 5.22 and 5.23, where despite the very strong approximation of closing ions energy equation with an isothermal closure, ions density, velocity, electric field and potential show all features very close to those presented in other literature results [6] [1] [7] [8]. Beside the already mentioned shapes



	$CFL_N$	$CFL_{el}$	Residual
Initial	100	100	-5
Restarted	50	100	-6

Table 5.6: Restart settings for Hall-Thruster simulation with magnetic field

of electric field and ions density, also ions velocity profile is reproduced in a way to accelerate up to  $17 - 18[km/s]$  [8] and electrostatic potential drops almost linearly close to the cathode region. As far as DTS performances are concerned, test case (*Case 3*) was useful to test the efficiency of a *restarted* simulation. Numerical settings introduced in section 5.3 are changed once a NaN is found within a DTS sub iteration: simulation is then restarted (in this specific test case through a *static* restart i.e. reading parameters from input file) with "better" performing parameters. As shown in Tab. 5.6 restarting the simulation by decreasing the initial  $CFL_N$  down to 50 and the residual to  $-6$ , results in a computation 18 times faster than an explicit one (see Tab. 5.7).

Method	Physical time [s]	Computational time	$CFL_{el}$
Explicit	$2.534 \times 10^{-6}$	1h 14m	0.9
DTS	$2.534 \times 10^{-6}$	4m	100

Table 5.7: Time performances for restarted DTS in Hall-Thruster test case without magnetic field

# Chapter 6

## Conclusions

In this thesis project a 1D multi-fluid solver for ions and electrons population simulation has been developed. It was first verified through very standard testcase and secondly used in a bit more complex cases where the numerical framework was not trivial. Some good results were obtained especially as far as the computational speed-up thanks to the DTS implicit scheme is concerned. Eventually this approach was used to try to estimate some parameters and features of an Hall-Thruster engine obtaining only some preliminary and mostly not precise results. Despite this one has to take into account that being able to properly model physics of electrons and reproducing in a trustworthy way their dynamics is all but an easy task: starting from the very big approximation that electrons always distribute according to a Maxwellian, when it has been proven that for high temperature they rather follow a bi-modal distribution function. Second big thing not to forget is that modeling plasma with a fluid approach is indeed easier under the implementation point of view, with respect for example to a PIC method with collisional terms studied using Montecarlo method, but unfortunately it is a further approximation of every Lagrangian-like approach like the latter, and moreover all the issues connected with numerics arises when dealing with a CFD code.

## 6.1 Future works

Hall-Thruster devices are characterized by a complex and three dimensional physics: within this thesis work only 1D features of the engine are studied. Moreover not only a dynamics for neutrals must be included and combined with those of ions and electrons if phenomena like breathing modes want to be studied, but also a proper modeling for heat flux closure and stress tensor contribution have to be considered. The lack of all these features represent a great hinder for a model when it has to deal with more complex dynamics than those of a collisionless plasma expansion into vacuum. A further development might have to do with including further moments closures in order to improve the precision of the model and range of applicability. If one wants to do that it is essential to switch to a Object-Oriented-Programming technique that allows a way greater versatility and code efficiency: even if in fact a FORTRAN Function-Oriented-Programming is probably fine up to the very first standard test case, it makes a cumbersome challenge introducing modification once one wants to include more features in the code.

# Bibliography

- [1] E. Ahedo. “Model of the plasma discharge in a Hall thruster with heat conduction”. In: *Physics of Plasmas* 9 (2002).
- [2] E Anderson. *LAPACK: Linear Algebra PACKage*. URL: <http://www.netlib.org/lapack>.
- [3] M. Benilov. “A kinetic derivation of multifluid equations for multi-species nonequilibrium mixtures of reacting gases”. In: *Physics of Plasmas* (4) (1997).
- [4] M. Benilov. “Momentum and energy exchange between species of a multicomponent gas mixture due to inelastic and reactive collisions”. In: *Physics of Plasmas* (3) (1996).
- [5] J. Blazek. *Computational Fluid Dynamics: applications and principles*. Elsevier, 2015.
- [6] S. Boccelli. “Collisionless ion modeling in Hall thrusters: Analytical axial velocity distribution function and heat flux closures”. In: *Phys. Plasma* 27 (2020).
- [7] J.P. Boeuf. “Low frequency oscillations in a stationary plasma thruster”. In: *J. Appl. Phys.* 84, 3541 (1998).
- [8] J.P. Boeuf. “Tutorial: Physics and modeling of Hall thrusters”. In: *Journal of Applied Physics* (2017).
- [9] H.M. Buecker. *On CFL Evolution Strategies for Implicit Upwind Methods in Linearized Euler Equations*. Tech. rep. Institute for Scientific Computing, RWTH Aachen University, D-52056 Aachen, Germany, 2006.
- [10] R. Burden. *Numerical Analysis*. Richard Stratton, 2010.

- [11] F. Chen. *Plasma Physics and controlled fusion*. Plenum Press, 1984.
- [12] J. E. Dennis. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, 1983.
- [13] M. Goebel. “Fundamental of Electric Propulsion: Ion and Hall Thrusters”. In: *NASA report* (2008).
- [14] G. Hagelaar. “Modelling methods for low-temperature plasmas”. PhD thesis. Laboratoire Plasma et Conversion d’Energie (LAPLACE) Université Paul Sabatier, 2008.
- [15] C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Birkhaeuser, 2007.
- [16] A. Kuppermann. *Chemical Reaction Cross Sections and Rate Constants*. Tech. rep. Gates and Creliin Laboratories of Chemistry2 California Institute of Technology Pasadena, 1968.
- [17] Randall J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhaeuser, 1999.
- [18] LXCAT. URL: <https://us.lxcat.net/home/>.
- [19] P. Mora. “Plasma expansion into vacuum”. In: *The American Physical Society, Volume 90* (2003).
- [20] A. Nishida. *LIS User Guide*. URL: <http://www.ssisc.org/>.
- [21] T.H. Pulliam. *Time accuracy and the use of implicit methods*. Tech. rep. Associate Fellow AIAA NASA Ames Research Center, 2015.
- [22] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1999.
- [23] S. Venkateswaran. “Analysis of preconditioning methods for Euler and Navier-Stokes equations”. In: *30th Computational Fluid-Dynamics (von Karman Institute of Technology)*. 1999.

# Appendices

## .1 Source term derivation

### INELASTIC COLLISIONS

Full thorough derivation from kinetic theory is presented in article [4], here will be only presented the final results and assumptions made within Benilov's model in order to build the presented closure for source terms. Assuming the the following reaction:



the rate of change of the number density of a species  $\gamma$  due to production of the particles of this species (**source term for continuity equation**  $\rho_\alpha w_\gamma = \rho_\alpha \nu_{\alpha\beta}$ ) is equal to:

$$w_\gamma = \nu_{\alpha\beta}$$

where indeed the derivation of each collisional frequency term has been discussed with the section dedicated to reaction rates. In order to calculate quantities  $\mathbf{w}_\gamma^{(m)}$  and  $w_\gamma^{(e)}$ , i.e., the rates of change of momentum (**source term for momentum equation**  $\rho_\alpha \mathbf{w}_\gamma^{(m)} = \mathbf{R}_{\gamma,inelastic}$ ) and energy of a species (**source term for energy equation**  $\rho_\alpha w_\gamma^{(e)} = S_{\gamma,inelastic}$ ) due to production of the particles of this species in a reaction, one needs to consider a term of the kinetic equation for the species  $\gamma$  accounting for this production.

$$\begin{aligned} \frac{\mathbf{w}_\gamma^{(m)}}{\mu_\gamma \nu_{\alpha\beta}} = & m_\alpha \mathbf{v}_\alpha + m_\beta \mathbf{v}_\beta - \left[ \left( \xi_{\alpha\beta}^{(m)} - 1 \right) \frac{T_\alpha - T_\beta}{T_{\alpha\beta}} m_{\alpha\beta} \right. \\ & \left. + \tilde{\xi}_{\alpha\beta}^{(m)} \sqrt{B} m_\delta \right] \mathbf{v}_{\beta\alpha} \end{aligned}$$

One can conclude that the momentum and energy lost by reacting species are divided between the species products proportionally to their particle masses. The second term on the right-hand side of *rate-of-momentum-change* equation appears due to anisotropy of the reaction cross section and describes change of momentum of one of the species-products at the expense of another. The second term on the right-hand side of *rate-of-energy-change* equation is due to the energy production (or consumption) in the reaction.

$$\begin{aligned}
\frac{w_\gamma^{(e)}}{\mu_\gamma \nu_{\alpha\beta}} &= \frac{3k (T_\alpha + T_\beta)}{2} + \frac{m_\alpha v_\alpha^2 + m_\beta v_\beta^2}{2} \\
&+ (\eta_{\alpha\beta} - 1) \frac{3k (\mu_\alpha T_\beta^2 + \mu_\beta T_\alpha^2)}{2T_{\alpha\beta}} \\
&- \left( \xi_{\alpha\beta}^{(m)} - 1 \right) m_{\alpha\beta} \mathbf{v}_{\beta\alpha} \cdot \frac{T_\alpha \mathbf{v}_\alpha - T_\beta \mathbf{v}_\beta}{T_{\alpha\beta}} + \frac{m_\delta}{m_\gamma} E \\
&- \xi_{\alpha\beta}^{(e)} \frac{m_\gamma - m_\delta}{m_\gamma} \frac{3k T_{\alpha\beta}}{2} \\
&- \tilde{\xi}_{\alpha\beta}^{(m)} \sqrt{B} m_\delta \mathbf{v}_{\beta\alpha} \cdot \frac{\mu_\alpha T_\beta \mathbf{v}_\alpha + \mu_\beta T_\alpha \mathbf{v}_\beta}{T_{\alpha\beta}} \\
&+ \tilde{\xi}_{\alpha\beta}^{(e)} \sqrt{B} \mu_\delta 3k (T_\alpha - T_\beta)
\end{aligned}$$

One can see that the fraction of this energy transferred to (or taken from) each species-product is proportional to the particle mass of another species product. The third and the fourth terms describe the increase of energy of one of the species products at the expense of another due to, respectively, imparity of particle masses of the products (the energy of a heavier species decreases and of a lighter one increases) and due to anisotropy of the reaction cross section.

As far as **electrons** are concerned ( $\alpha$  and  $\gamma$  terms), for the *rate-of-momentum-change* equation differential cross section of the reaction is considered to be isotropic, and indeed electrons mass is much smaller than the neutrals one ( $\beta$  term), therefore no source term survives. Analogous reasoning are made for the *rate-of-energy-change* equation, for which only the second term indeed survives, and it is the one accounting for energy loss by electrons population within ionization and excitation type of collisions.

For what concerns **ions** instead (only  $\gamma$  terms), same hypothesis for the isotropicity of differential cross-section holds, being the electrons momentum much smaller than the one connected to the one of neutrals population, the source term is not void, but, as already discussed within the collisional source term section, it accounts for ions with same momentum features of pre-ionization neutrals. Following analogous reasoning one can show that only the first term of the *rate-of-energy-change* equation for ions populations is surviving.



## ELASTIC COLLISIONS

Full thorough derivation is carried out in [3], here will be only presented the final results and assumptions made within Benilov's model in order to build the presented closure for source terms. The rate of momentum exchange due to elastic collision may be written as:

$$\mathbf{R}_{\alpha,elastic} = \mathbf{r}_{\alpha\beta}^{(m)} = \frac{n_{\alpha}n_{\beta}kT_{\alpha\beta}}{nD_{\alpha\beta}} (\mathbf{v}_{\alpha} - \mathbf{v}_{\beta})$$

that results into the contribution already derived for the elastic collisional source term of ions and electrons population, proportional to the difference between neutrals mean velocity (negligible) and electrons (or ions) mean velocity. The rate of energy exchange due to elastic collision may be written as:

$$\begin{aligned} S_{\alpha,elastic} = r_{\alpha\beta}^{(e)} = & \frac{n_{\alpha}n_{\beta}k}{nD_{\alpha\beta} (m_{\alpha} + m_{\beta})} \\ & \times \left[ 3s_{\alpha\beta}kT_{\alpha\beta} (T_{\alpha} - T_{\beta}) + (m_{\alpha}T_{\beta}\mathbf{v}_{\alpha} + m_{\beta}T_{\alpha}\mathbf{v}_{\beta}) \right. \\ & \left. \cdot (\mathbf{v}_{\alpha} - \mathbf{v}_{\beta}) \right] \end{aligned}$$

which can be further simplified into two different contributions for electrons: one connected to kinetic and the other one to internal energy source (2nd and 1st terms respectively of equation here below)

$$r_{e,Xe}^{(e)} = 3 \frac{v_{el}m_en_ek(T_e - T_{Xe})}{m_{Xe}} \left( -\frac{Q(e)}{Q(m)} \right) + n_e v_{el} \left( \frac{m_e T_{Xe}}{m_{Xe} T_e} v_e + v_{Xe} \right) (v_e)$$

in which both contributions happen to be void due to the already mentioned very low electrons-to-neutrals mass ratio, and also to the fact that electrons temperature is much higher than that of neutrals. For the ions population, within this thesis framework, the approximation of avoiding energy exchange contribution was chosen, mainly due to its much more negligible contribution to the one that the electric field bring instead about.

## .2 LIS library

LIS library [20] is a C language based library written for solving linear system though iterative methods which moreover allows clever sparse matrix storage like the tridiagonal block matrix storage format that will be presented in Fig. 1 (BSR or Block Sparse Row).

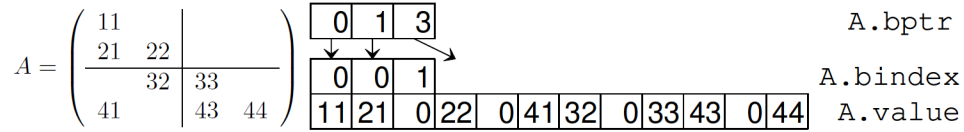


Figure 1: Data structure of BSR

where *bindex* is an integer array which stores the block column numbers of the nonzero blocks, *bptr* is an integer array which stores the starting points of the block rows in the array *bindex*, and *value* is a double precision array which stores the values of the elements of the nonzero blocks. LIS provides a useful FORTRAN interface, which can be used exclusively including preprocessing directive (*include "lisf.h"*) that allows to bind the two codes interfaces. Initialization part is as follows within *initialization.f90* file:

```
bnnz = 3*(N_cells-2) - 2

allocate (bptr(0:N_cells-2))
allocate (bindex(0:bnnz-1))
allocate (value_vector(0:bnnz*16-1))

! Building bptr structure
bptr(0) = 0
bptr(1) = bptr(0) + 2

do i=2,N_cells-3

    bptr(i) = bptr(i-1) + 3
```

```

end do

bptr(N_cells - 2)= bptr(N_cells - 3) + 2

! Building bindex structure

bindex(0) = 0
bindex(1) = 1

triple_index_iii = 0

do i=2,bnnz-3

    bindex(i) = triple_index_iii

    triple_index_iii = triple_index_iii + 1

    if (mod(i-1,3) .eq. 0) THEN

        triple_index_iii = triple_index_iii - 2

    end if

end do

bindex(bnnz-2) = N_cells-4
bindex(bnnz-1) = N_cells-3

allocate (LS_rhs((N_cells-2)*4))
allocate (Q_n((N_cells-2)*4))
allocate (Q_n_minus_1((N_cells-2)*4))

bnnz = 3*(N_cells-2) - 2
n_size = (N_cells-2) * 4
bnr = 4
bnc = 4

```

```

nr =(n_size-1)/bnr+1

! Initializaing matrix
call lis_matrix_create(comm,A_MATRIX,ierr)
call lis_matrix_set_size(A_MATRIX,0,n_size,ierr)

! Initializing solution
call lis_vector_duplicate(A_MATRIX,x_solution,ierr)
call lis_vector_duplicate(A_MATRIX,IMPLICIT_RHS,ierr)


! Initializing solver
call lis_solver_create(solver,ierr)
call lis_precon_create(solver,my_prec,ierr)
call lis_solver_set_option('-i 9 -p 2 -maxiter 100',solver,ierr)

whereas solver part is included in implicit-electrons-utilities.f90:

! Assembling main matrix
call lis_matrix_set_bsr(bnr,bnc,bnnz,bptr,bindex,
value_vector,A_MATRIX,ierr)
call lis_matrix_assemble(A_MATRIX,ierr)

do local_index=1,n_size

    call lis_vector_set_value(LIS_INS_VALUE,
    local_index, LS_rhs(local_index), IMPLICIT_RHS, ierr)

end do

! Creating solver
call lis_solve_kernel(A_MATRIX,IMPLICIT_RHS,
x_solution,solver,my_prec,ierr)

```

```

do  iii=1,n_size

    row_DC_index = mod(iii-1,4) + 1
    col_DC_index = (iii-1)/4 + 1

    call lis_vector_get_value(x_solution, iii,
    delta_cons(row_DC_index, col_DC_index), ierr)

end do

```

### .3 Predictor Corrector

One of the features introduced within the software *FORMICA* is the possibility to try and predict the electric field at next time iteration (n+1) using an approach coming from [14] that uses values of number densities and electric field at previous timesteps to predict the one at the following one. This feature is exclusively implemented withing the code with the purpose of improving the quality of the Jacobian tensor used within an implicit scheme, and hence improving the predicting power of this latter one. To be noticed is that the scheme proposed by [14] comes from a different mathematical modeling approach of plasma, based on continuity equations and drift-diffusion equations for ions and electrons coupled with Poisson's solver. The formulation to predict  $\mathbf{E}^{n+1}$  is the following:

$$-\nabla \cdot \left( \varepsilon_0 (1 + \chi_e) \nabla \Phi^{k+1} \right) = \sum_j q_j \left( 2n_j^k - n_j^{k-1} \right) - \nabla \cdot \left( \varepsilon_0 \chi_e \nabla \Phi^k \right)$$

with

$$\chi_e = \frac{\Delta t}{\varepsilon_0} \sum_j |q_j| \mu_j n_j^k = \frac{\Delta t}{\tau_d}$$

whose full implementation algorithm is presented here below:

$$\begin{aligned}
& (1 + \chi_{e,j}) \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{dx^2} + \\
& (\nabla \chi_{e,j}) \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2dx} = \\
& - \frac{1}{\varepsilon_0} \left( \sum q_i \left( 2n_{i,j}^k - n_{i,j}^{k-1} \right) + \right. \\
& \left. \varepsilon_0 \left( \nabla x_{e,j} E^k + x_{e,j} \frac{\rho^{Q,n}}{\varepsilon_0} \right) \right)
\end{aligned}$$

## .4 LAPACK library

LAPACK [2] is a Fortran language based library for solving linear system through direct methods. It is indeed very powerful when the size of the matrix is small, but especially when the matrix within the linear system has such a shape that a super-simplified algorithm for the resolution can be chosen. This is the case for the Poisson solver, for which Neumann and Dirichlet boundary conditions made it such that the shape of the Poisson's matrix is always tridiagonal: Thomas algorithm is in fact known to be a most suitable choice when solving tridiagonal linear systems, allowing a great space savings in memory during matrix's elements storage. Utilities used within software *FORMICA* are the following: at first those used to initialize matrices only once at the beginning of the simulation (including the matrix assembling on case of periodic boundary conditions)

```

if (BCs_PERIODIC_BOOLEAN) then

    ! Common Formulation tridiagonal matrix with values on
    interfaces only of phi-formulation
    with tridiagonality broken

    N = N_int
    KL = N-1
    KU = N-1
    LDAB = 2*KL + KU + 1
    allocate (AB(LDAB,N) , IPIV(N) , D(N) , DL(N-1) , DU(N-1))

```

```

DO i = 1, N-1

    D(i) = -alfa(i-1)*(1/l_cells(i) + 1/l_cells(i+1))
    DU(i) = alfa(i-1)/l_cells(i+1)
    DL(i) = alfa(i)/l_cells(i+1)

END DO

D(N) = -alfa(N-1)*(1/l_cells(N) + 1/l_cells(N+1))

! Formulation with tridiagonality broken

AB = 0

DO i = 1, N-1

    AB(KL+KU, i+1) = DU(i)
    AB(KL+KU+1, i) = D(i)
    AB(KL+KU+2, i) = DL(i)

end do

AB(KL+KU+1, N) = -1
AB(KL+KU+2, N) = 1
AB(KL+KU+2-N, N) = D(N)
AB(KL+KU+2-(N-1), N-1) = DL(N-1)
AB(KL+KU+2, N-1) = 0

else if ((BCs_DIR_BOOL_Right_EM .and. BCs_NEU_BOOL_Left_EM) .or.
(BCs_DIR_BOOL_Left_EM .and. BCs_NEU_BOOL_Right_EM)) then

N = N_int
allocate (D(N), DL(N-1), DU(N-1))

DO i = 1, N-1

```

```

        D(i) = -alfa(i-1)*(1/l_cells(i) + 1/l_cells(i+1))
        DU(i) = alfa(i-1)/l_cells(i+1)
        DL(i) = alfa(i)/l_cells(i+1)

    END DO

    D(N) = -alfa(N-1)*(1/l_cells(N) + 1/l_cells(N+1))

else ! Both Dirichlet conditions

    N = N_int - 2
    allocate (D(N),DL(N-1),DU(N-1))

    DO i = 1, N-1

        ! Note that l_cells(1) is the ghost_cell size ,
        ! so indexes are shifted by 1

        D(i) = -alfa(i)*(1/l_cells(i+1) + 1/l_cells(i+2))
        DU(i) = alfa(i)/l_cells(i+2)
        DL(i) = alfa(i+1)/l_cells(i+2)

    END DO

    D(N) = -alfa(N)*(1/l_cells(N+1) + 1/l_cells(N+2))

end if

    and then those used to solve the linear system:
    if (BCs.PERIODIC.BOOL_EM) then

        ! Common Formulation tridiagonal matrix with values
        ! on interfaces only of phi-formulation

```



```

with tridiagonality broken

N = N_int
allocate(B(N))
LDB = N
B = force_term_inter
B(1) = B(1) + B(N)
B(N) = 0
allocate(ABC(LDAB,N))
ABC = AB

! N is order of Matrix, in our case has to be number
of interfaces (N_int)
! NRHS is the number of column of RHS, indeed 1
! D contains elements of diagonal of A
! B is the RHS, which will come out as solution
! LDB dimension of array B
! KL subdiag
! KU upperdiag
! ABC has matrix A stored somehow inside
! IPIV pivoting array

call dgbsv (N, KL, KU, NRHS, ABC, LDAB, IPIV, B, LDB, INFO)

else if (BCs_DIR_BOOL_Left_EM .and. BCs_NEU_BOOL_Right_EM) then

!!!! Neumann on the right – Dirichlet on the left

N = N_int
allocate(D_(N),DL_(N-1),DU_(N-1),B(N))
B = force_term_inter(2:N_int-1)

LDB = N
DL_ = DL
D_ = D
DU_ = DU

```

```

! Neumann Right - Dirichlet Left
B(1) = phi_0
B(N) = Neu_right * l_cells(N)

D(1) = 1.00d0
DU(1) = 0.0d0
D(N) = -1.0d0
DL(N-1) = 1.0d0

! N is order of Matrix, in our case has to be number
of internal interfaces (N_cells -3 )
! NRHS is the number of column of RHS, indeed 1
! DL contains elements of lower-diagonal of A
! DU contains elements of upper-diagonal of A
! D contains elements of diagonal of A
! B is the RHS, which will come out as solution
! LDB dimension of array B

CALL dgtsv( N, NRHS, DL_, D_, DU_, B, LDB, INFO )

else if (BCs_DIR_BOOL_Right_EM .and. BCs_NEU_BOOL_Left_EM) then

!!!! Neumann on the left - Dirichlet on the right

N = N_int
allocate(D_(N),DL_(N-1),DU_(N-1),B(N))
B = force_term_inter(2:N_int-1)

LDB = N
DL_ = DL
D_ = D
DU_ = DU

! Neumann Left - Dirichlet Right
B(1) = Neu_left * l_cells(2)

```

```

B(N) = phi_end

D(1) = 1.00d0
DU(1) = -1.0d0
D(N) = 1.00d0
DL(N-1) = 0.000d000

! N is order of Matrix, in our case has
to be number of internal interfaces (N_cells -3 )
! NRHS is the number of column of RHS, indeed 1
! DL contains elements of lower-diagonal of A
! DU contains elements of upper-diagonal of A
! D contains elements of diagonal of A
! B is the RHS, which will come out as solution
! LDB dimension of array B

CALL dgtsv( N, NRHS, DL_, D_, DU_, B, LDB, INFO )

else

N = N_int-2
allocate (D_(N),DL_(N-1),DU_(N-1),B(N))
B = force_term_inter(2:N_int-1)
B(1) = B(1) - alfa(1)/l_cells(2)*phi_0
B(N) = B(N) - alfa(N)/l_cells(N+2)*phi_end

LDB = N
DL_ = DL
D_ = D
DU_ = DU

! N is order of Matrix, in our case has
to be number of internal interfaces (N_cells -3 )
! NRHS is the number of column of RHS, indeed 1
! DL contains elements of lower-diagonal of A

```

```

! DU contains elements of upper-diagonal of A
! D contains elements of diagonal of A
! B is the RHS, which will come out as solution
! LDB dimension of array B

CALL dgtsv( N, NRHS, DL_, D_, DU_, B, LDB, INFO )

end if

```