



# Models, manual and validations for FENNECS code

---

Le Bars Guillaume

*Supervisors :*

MER. Jean-Philippe Hogge

Dr. Joaquim Loizu

MER. Stefano Alberti

September 29, 2022

# Contents

<b>1</b>	<b>The problem</b>	<b>3</b>
<b>2</b>	<b>FENNECS a 2D electrostatic Particle in cell code</b>	<b>5</b>
2.1	Numerical model . . . . .	5
2.2	Normalisations . . . . .	6
2.3	Input parameters . . . . .	7
<b>3</b>	<b>Code validation</b>	<b>15</b>
3.1	Particle loading . . . . .	19
3.2	Physical quantities at the end of the simulation . . . . .	24
3.3	Particles trajectories . . . . .	31
3.4	Time evolutions and spectrum . . . . .	31
<b>4</b>	<b>Finite element method: weighted B-splines</b>	<b>35</b>
4.0.1	Univariate and multi-variate b-splines . . . . .	36
4.0.2	Representation of the inhomogeneous Dirichlet boundary conditions (transfinite interpolation) . . . . .	38
4.0.3	More complex test case . . . . .	38
<b>5</b>	<b>Parallelisation</b>	<b>40</b>
5.1	Space decomposition . . . . .	40
5.2	Implementation . . . . .	41
5.3	Measured scaling and improvements . . . . .	42
<b>6</b>	<b>Azimuthal rotation frequency study</b>	<b>44</b>
6.1	Davidson stable force balance . . . . .	44
<b>7</b>	<b>Characteristic time scales</b>	<b>49</b>
<b>8</b>	<b>Maximum density estimate</b>	<b>50</b>
8.1	Case of an annular electron layer in a coaxial geometry with magnetic mirror . . . . .	51
<b>9</b>	<b>Estimated gained energy for particles created with 0 kinetic energy</b>	<b>52</b>
<b>10</b>	<b>Diocotron Modes</b>	<b>53</b>
10.1	Case of axial uniformity (infinite length coaxial configuration) . . . .	53

<i>CONTENTS</i>	3
11 Ionisations simulations using PIC-MCC	54
12 Experimental diagnostics	55
13 Future investigations	56
A Conservation of $H$ and $P_\theta$ in coaxial geometry	57
B Self consistent $\phi$ in coaxial geometry	59

# Chapter 1

## The problem

The code FENNECS developed at SPC simulates a collisionless non-neutral plasma contained in a coaxial geometry of finite length  $2L$  as shown in figure 1.1. It has been developed to study the behaviour of trapped electrons close to the electron gun in a gyrotron. The coaxial insert simulates the electron gun while the outer cylinder simulates the gyrotron control electrode. Due to the azimuthal symmetry about the  $\hat{z}$  axis ( $\frac{\partial}{\partial\theta} = 0$ ), the simulation is restrained to the  $(r, z)$  plane. In the electrostatic limit, the dynamic of this plasma can be described by the Vlasov-Poisson equations:

$$\left( \frac{\partial}{\partial t} + \mathbf{v} \cdot \frac{\partial}{\partial \mathbf{x}} + \frac{e}{m_e} (\mathbf{E} + \mathbf{v} \times \mathbf{B}_0) \cdot \frac{\partial}{\partial \mathbf{v}} \right) f(t, \mathbf{x}, \mathbf{v}) = 0, \quad (1.1)$$

$$\nabla^2 \Phi(t, \mathbf{x}) = -\frac{e}{\epsilon_0} \int f(t, \mathbf{x}, \mathbf{v}) d^3 \mathbf{v}. \quad (1.2)$$

Here  $e$  is the electron charge;  $m_e$  its mass;  $\mathbf{x}$  and  $\mathbf{v}$  are the position and velocity;  $\epsilon_0$  is the electric permittivity of vacuum;  $f$  is the electrons distribution function;  $\Phi$  is the electric potential;  $\mathbf{E} = -\nabla \Phi$  is the self-consistent electric field;  $B_0$  is external magnetic field.

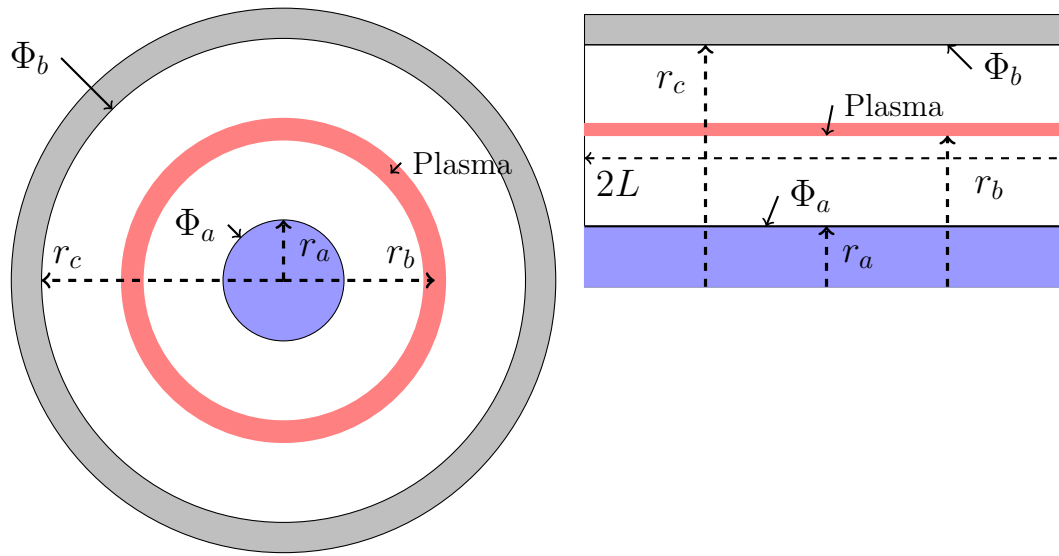


Figure 1.1: Geometry of the simulated problem seen in the  $(x, y)$  and  $(r, z)$  plane. To simulate the system without the coaxial insert,  $r_a$  is set to 0.

# Chapter 2

## FENNECS a 2D electrostatic Particle in cell code

### 2.1 Numerical model

To solve the Vlasov-Poisson system of equations 1.1 and 1.2, the distribution function  $f$  is sampled by  $N_p$  macro-particles simulated by FENNECS. The distribution function is in that case written as a sum of point charges:

$$f(t, \mathbf{x}, \mathbf{v}) = \sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}_i) \delta(\mathbf{v} - \mathbf{v}_i). \quad (2.1)$$

The evolution of the distribution function therefore follow the evolution of the position and velocity of the electrons. These quantities will be obtained by solving the electrons equations of motion:

$$\frac{d\mathbf{p}_i(t)}{dt} = -e(\mathbf{E}(r, z, t) + \mathbf{v}_i(t) \times \mathbf{B}_0(r, z)), \quad (2.2)$$

$$\frac{d\mathbf{r}_i(t)}{dt} = \frac{\mathbf{p}_i(t)}{m_e \gamma_i}, \quad (2.3)$$

using the self-consistent electric field  $\mathbf{E}(r, z) = -\nabla\Phi(r, z)$  obtained through the resolution of Poisson's equation. With

$$\mathbf{p}_i = m_e \gamma_i \mathbf{v}_i, \quad (2.4)$$

the  $i^{th}$  electron momentum,  $v_i$  its velocity and  $\gamma_i = \frac{1}{\sqrt{1-v_i^2/c^2}}$  its lorentz factor. Here  $c$  is the velocity of light in vacuum. The electric potential boundary conditions are in  $z$

$$\left. \frac{\partial \Phi}{\partial z} \right|_{z=-L} = 0; \quad \left. \frac{\partial \Phi}{\partial z} \right|_{z=L} = 0; \quad (2.5)$$

and in  $r$  for  $r_a \neq 0$

$$\Phi(r_a) = \Phi_a; \quad \Phi(r_b) = \Phi_b. \quad (2.6)$$

For code validation purposes another boundary condition has also been used when the coaxial insert radius is set to  $r_a = 0$

$$\left. \frac{\partial \Phi}{\partial r} \right|_{r=0} = 0; \quad \Phi(r_b) = \Phi_b. \quad (2.7)$$

To solve Poisson's equation, the charge density  $\rho$  of the plasma is defined as

$$\rho(\mathbf{r}) = -e \sum_{i=1}^{N_p} \frac{\delta(r - r_i) \delta(z - z_i)}{2\pi r}. \quad (2.8)$$

The particle boundary conditions are periodic which means that if a particle escapes at the right boundary, it will be injected back with the same  $r$  position and same velocity at the left boundary.

## 2.2 Normalisations

For better numerical accuracy, all physical variables are normalised in the code. This imposes that physical quantities are close to 1 in the code. The other advantage of these normalisations is that the risk of variable overflow is greatly reduced. In this section, we will present the Normalisations used in the code. The normalisation constant in time is

$$t_{norm} = \frac{1}{\omega_{ce}}, \quad (2.9)$$

and in velocity

$$v_{norm} = c. \quad (2.10)$$

The normalisation used for distances is

$$r_{norm} = v_{norm} t_{norm} = \frac{c}{\omega_{ce}}; \quad (2.11)$$

for magnetic fields

$$B_{norm} = B_{Max}; \quad (2.12)$$

for electric fields

$$E_{norm} = v_{norm} B_{norm} = c B_{Max}; \quad (2.13)$$

and for the electric potential

$$\Phi_{norm} = E_{norm} r_{norm} = \frac{B_{Max} c^2}{\omega_{ce}}. \quad (2.14)$$

With  $\omega_{ce} = \frac{e B_{Max}}{m_e}$  the cyclotron frequency,  $c$  the speed of light in vacuum,  $B_{Max}$  the maximum value of the magnetic field in the simulated space,  $e$  the electron charge and  $m_e$  its mass. For the non-ideal power supply module the following normalisations have been added. The electric charge normalisation

$$q_{norm} = ew; \quad (2.15)$$

with  $w$  the macro-particle weight. The resistance normalisation

$$R_{norm} = t_{norm}\phi_{norm}/q_{norm} \quad (2.16)$$

and the Capacitance normalisation

$$C_{norm} = q_{norm}/\phi_{norm}. \quad (2.17)$$

## 2.3 Input parameters

To define the simulation parameters for a run of FENNECS, a single text file containing fortran namelists is given as a command line parameter to the executable. In this subsection is described the current namelist and its parameters:

### &BASIC

**Variable** (default value) Description

**dt** ( ) Time step expressed in seconds

**it0d** (1) Defines the number of steps between the saving of each scalar variable not depending on  $r$  and  $z$  ( $E_{pot}$ ,  $E_{kin}$  ...).

**it2d** (1) Defines the number of steps between each write to the hdf5 of the 2d variables( $\Phi, E, \dots$ ).

**itparts** (1) Defines the number of steps between each write to the hdf5 of the full particles position and velocity.

**ittracer** (1) Defines the number of steps between each write to the hdf5 of the particles position and velocity if they are defined with the tracer property.

**ittext** (1) Defines the number of steps between each write, to the program standard output, of the total potential energy, kinetic energy and error in the energy.

**nbcelldiag** (0) Defines the number of celldiagnostics groups in &celldiagparams.

**itcelldiag** (100000) Defines the number of steps between each write of the celldiag data.

**resfile** ('results.h5') Name and path of the hdf5 file containing the simulation result.

**nlres** (.FALSE.) Sets if this run is a restart. This means that during the program initialization, the particles position and velocities will be read from a "restart.h5" file, and the new simulation data will be appended to the existing **resfile**



**newres** (.FALSE.) Sets if the result file **resfile** is a new file(.TRUE.) and should be created in the case of a restart. This means that during the program initialization, the particles position and velocities will be read from a "restart.h5" file, and the new simulation data will be saved in the new **resfile**

**nlxg** (.FALSE.) Set if the graphical interface and plots should be displayed

**nlPhis** (.TRUE.) Set if the macro particles will interact through the self consistent Electric field. If set to .FALSE. the self consistent electric potential and fields will be evaluated and saved in the hdf5 result, but not applied to the particles. However, the external electric field is always applied.

**nlclassical** (.FALSE.) If set to .TRUE.,  $\gamma$  is set to 1 for every particle throughout the simulation. This is equivalent to solving the equations of motion classically.

**nplasma** () Number of macro-particles simulated.

**potinn** () Value of the electric potential on the surface of the inner cylinder( $\Phi(r_a)$ ).

**potout** () Value of the electric potential on the surface of the outer cylinder( $\Phi(r_c)$ ).

**radii** () 4 component array containing in order for the  $r$  dimension, the lower limit of the mesh, the first limit between the coarse mesh and the fine mesh, the second limit between the fine mesh and the coarse mesh, the upper limit of the mesh( $[r_a, r_f^-, r_f^+, r_b]$ ).

**lz** () 2 component array containing the lower and upper limit of the mesh( $[z^-, z^+]$ ).

**nz** () Number of intervals in  $z$ .

**nnr** () 3 component vector containing the number of intervals in  $r$  for the 3 mesh regions ( $[n_{coarse,1}, n_{fine}, n_{coarse,2}]$ ).

**ngauss** () 2 component array containing the number of gauss points used for the integration in the FEM method( $[ngauss_z, ngauss_r]$ ).

**femorder** () 2 component array containing the order of the B-splines functions used in the FEM method( $[order_z, order_r]$ ).

**distribtype** (1) Switch parameter defining the distribution function used to load the particles.

1. Uniform distribution in  $z$ ,  $1/r$  distribution in  $r$ , Gaussian distribution in each component of the velocity given the temperature "temp".
2. Stable distribution for a magnetic mirror as described in section 3 and in [?].
3. Same as 2 but with  $n_0(r, z) \propto 1/r^2$ .

- plasmadim** () 4 component array containing the  $z$  and  $r$  limits of the electron cloud used by **Distribtype**=1 at the initialization of the particles position( $[z_{min}, z_{max}, r_{min}, r_{max}]$ ).
- n0** () Initial value of the density factor used in the different **Distribtype**.
- B0** () Magnetic mirror scaling factor.
- Rcurv** () Magnetic mirror ratio.
- width** () Magnetic mirror width.
- temp** () Temperature expressed in Kelvins used for **Distribtype**=1.
- H0** () Initial value of the hamiltonian  $H_0$  used in **Distribtype**=2,3.
- P0** () Initial value of the Canonical angular momentum  $P_0$  used in **Distribtype**=2,3.
- nblock** () Number of slices in  $z$  used to approximate the electron cloud boundary for particle initialization in **Distribtype**=2,3.
- weights\_scale** (1.0) Allows to rescale the macro-particle weight of the main specie on restart.
- magnetfile** ("") Name of the h5 file containing the description of the magnetic field.
- bscaling** (-1) Defines the way the magnetic field is rescaled when read from hdf5 file. (0) No rescale, (1) Rescale the maximum value fo the magnetic field amplitude in the h5 file (could contain points outside of the simulation grid). (-1) Rescale the magnetic field amplitude after calculation of the field at the grid points.

### &celldiagparams

This defines the behavior of the cell diagnostic that stores the particles velocity and position for all particles in a given cell. The arguments are a list that allow you to define several of these diagnostics

**Variable** (default value) Description

- specieid** () List of species with index corresponding to the indices in partslist
- rindex** () radial index of the cell to consider
- zindex** () axial index of the cell to consider

**&geometry**

Namelist used to define the geometry using weighted-splines and allowing boundary conditions on curvi-linear domains. Currently Dirichlet boundary conditions are used on the metallic parts and Neumann are used otherwise. The center cylinder will be set at potential Potinn and the external cylinder and ellipse are set at Potout of Basic namelist

**Variable** (default value) Description

**walltype** (0) Type of configuration to consider

- \*. Verification case where a source-term is imposed and configuration \* is used for the geometry. With this, the correctness of the Poisson solution can be tested in any geometry. The manufactured electric potential solution has the form:  $\sin(\pi(z - z_c)/L_z) * \sin(\pi(r - r_c)/L_r) + 2$
- 0. Coaxial configuration of constant radius with central cylinder and external cylinder.
- 1. Center cylinder and external ellipse.
- 2. Center cylinder and combination of external cylinder with metallic ellipse
- 3. Two facing ellipses with extended cylinders.
- 9. Geometry read from a b-spline description of the boundaries and defined by the namelist &spldomain and an h5 file.

**nlweb** (.true.) Toggle if weighted-extended-b-splines (t) or simple weighted-bsplines (f) must be used. There is better numerical precision and stability if set to true.

**testkr** (1) For testing purposes defines the radial wavelength  $L_r = (r_{max} - r_{min})/testkr$  of the imposed source term.

**testkz** (1) For testing purposes defines the axial wavelength  $L_z = (z_{max} - z_{min})/testkz$  of the imposed source term.

**z\_0** () axial center of the ellipse

**r\_0** () radial center of the ellipse

**z\_r** () axial radius of the ellipse

**r\_r** () radial radius of the ellipse

**r\_a** () radius of the center metallic cylinder

**r\_b** () radius of the external metallic cylinder

**Interior** (-1) Defines if the inside or outside of the ellipse is considered in the geometry.

**above1** (1) Defines if the vacuum region is outside (1) or inside (-1) the cylinder of radius  $r\_a$ .

**above2** (-1) Defines if the vacuum region is outside (1) or inside (-1) the cylinder of radius  $r\_b$ .

### **&spldomain**

Defines the behavior of the splinebound module which allows to set boundaries using a b-spline curve representation of the metallic surfaces.

**Variable** (default value) Description

**dist\_extent** () Set the distance in m over which the geometric weight goes from 0 to 1 away from the boundary.

**h5fname** () name of the h5 file containing the boundary description.

**Dvals** () array storing the fixed potential in V for each boundary.

### **&maxwellsourceparams**

This defines the behavior of a volumetric source creating particles uniformly in the axial direction, according to a specified distribution in the radial direction and according to a Maxwellian distribution in velocity.

**Variable** (default value) Description

**frequency** () Number of macro-particles created per second of simulated time.

**temperature** () temperature in kelvin used in the Maxwellian distribution function.

**rlimits** () array storing the radial extent of the source.

**zlimits** () array storing the axial extent of the source.

**time\_start** (-1) time in seconds at which the source must be turned on. -1 means start from beginning of simulation

**time\_end** (-1) time in seconds at which the source must be turned off. -1 means, never turn off the source.

**radialtype** (2) type of radial distribution to use when creating particles

1.  $1/r$  distribution in  $r$ .
2. uniform distribution in  $r$ .
3.  $1/r^2$  distribution in  $r$ .
4. gaussian distribution in  $r$  centered at  $0.5(rlimits(1) + rlimits(2))$  and with  $\sigma = 0.1(rlimits(2) - rlimits(1))$ .

### **&partsload**

Defined in specific particles loading files, this namelist allows more flexibility than the legacy loader. It is also combined with either a list of particles to load or a list of slices and the number of macro-particles in each to load non-trivial distributions.

**Variable** (default value) Description

**partformat** ('slice') Type of particle file.

**'slice'** the cloud is defined spacially as a list of slices with on each line, the left axial limit, the lower and upper radial limit and the number of macro-particles to load in this slice. The right limit is given on the next line and the last line comports only the axial limit.

**'parts'** the cloud is defined by individual particles with in order: the radial, azimuthal and axial position, then the radial, azimuthal and axial velocity in si units.

**nblock** () number of slices in the slice description

**mass** (m\_e) the mass of a physical particle in kg.

**charge** (q\_e) the charge of a physical particle in C.

**weight** (1) the number of physical particles that one macro-particle represents.

**npartsalloc** () size of the initial particles array to prepare for increase of particle numbers.

**radialtype** (2) type of radial distribution to use when creating particles

1.  $1/r$  distribution in  $r$ .
2. uniform distribution in  $r$ .
3.  $1/r^2$  distribution in  $r$ .
4. gaussian distribution in  $r$  centered at  $0.5(rlimits(1) + rlimits(2))$  and with  $\sigma = 0.1(rlimits(2) - rlimits(1))$ .

**velocitytype** (1) type of velocity distribution to use when creating particles

1. Maxwellian velocity of 0 mean and temperature defined in temperature.
2. Davidson stable distribution defined with H0 and P0
3. flat top distribution with mean meanv and span spanv

**temperature** (10000) temperature in kelvin used for the Maxwellian velocity distribution function.

**H0** ( $3.2 \cdot 10^{-14}$ ) Energy in joules of the particles for Davidson distribution function.

**P0** ( $8.66 \cdot 10^{-25}$ ) Canonical angular momentum in si of the particles for Davidson distribution function.

**is\_test** (.false.) if true the specie is only a test specie for the trajectories and will not participate in the calculation of the self-consistent electric field.

**meanv** (0,0,0) mean velocity in radial, azimuthal and axial direction for the flat-top velocity loading.

**spanv** (0,0,0) span of the flat-top velocity in radial, azimuthal and axial direction for the flat-top velocity loading. Velocities will be contained between meanv-spanv and meanv+spanv.

### **&neutcolparams**

This defines the behavior of the elastic and ionisation collision between electrons and residual neutral gas.

**Variable** (default value) Description

**nlcol** (.false.) defines if the collision are active or not. If both `ela_cross_sec_file` and `io_cross_sec_file` are empty, the collisions are deactivated.

**neutdens** ( $2.4 \cdot 10^{16} m^{-3}$ ) density of neutral particles per cubic meters.

**Eion** (21.56 eV) ionization energy of the neutral considered.

**scatter\_fac** (24.2 eV) tabulated scatter factor used to compute the fraction of energy between scattered and created electrons in a ionization event.

**io\_cross\_sec\_file** () name of the file containing the table of cross-sections as a function of energy in eV for the ionisation. In this file the comments are indicated with "!" and the table is just a two column list with the energy in eV and the cross-section in square meter.

**ela\_cross\_sec\_file** () name of the file containing the table of cross-sections as a function of energy in eV for the elastic collisions. In this file the comments are indicated with "!" and the table is just a two column list with the energy in eV and the cross-section in square meter.

**species** (1,-1) The first number sets the colliding specie. In the case of ionisation, the second number defines in which specie the ions should be added with 0 velocity. If the second number is lower than 1, no ions are created.

### **&psupplyparams**

Sets the parameters of the non-ideal power supply.

**Variable** (default value) Description

**expneutdens** () neutral density, in  $m^{-3}$ , measured in the experiment we want to simulate. This permits correct time-scale separation and rescaling because we accelerate the ionisation time-scales.

**PsResistor** () Internal resistance of the power supply in Ohms.

**geomcapacitor** () Capacitance of the geometry in Farrads.

**targetbias** () Bias in V set on the power supply.

**nbhdt** () Number of Boris algorithm time steps between each half time-steps of the runge kutta algorithm used to compute the time-evolution of the bias at the surface of the electrodes.

**active** (.false.) Defines if this module is active.

# Chapter 3

## Code validation

The validation of the code is first done by loading the system using a known stable distribution function described in [?]. This problem assumes a finite length cylinder in a magnetic field topology set as a magnetic mirror as shown in figures 3.1 and 3.2. The plasma is a tenuous non-relativistic electron layer confined axially and radially by the magnetic mirror field.

The electron distribution function is given by:

$$f(r, z, \mathbf{p}) = \frac{\hat{n}_e \hat{r}_b^-}{2\pi m_e} \delta(H - H_0) \delta(P_\theta - P_0), \quad (3.1)$$

with

$$H = \frac{1}{2m_e}(\mathbf{p}^2) - e\Phi_0(r, z), \quad (3.2)$$

$$P_\theta = r(p_\theta - eA_\theta^{ext}(r, z)). \quad (3.3)$$

Here,  $H$  is the total energy;  $P_\theta$  is the canonical angular momentum;  $e$  is the electron charge;  $\mathbf{p}$  is the electron momentum;  $\Phi_0$  is the self-consistent electric potential;  $\hat{r}_b^-$  is the inner boundary of the electron layer;  $\hat{n}_e$  is the electron density at radius  $\hat{r}_b^-$ ;  $H_0$  and  $P_0$  are positive constants.

The magnetic vector potential  $A_\theta^{ext}$  is approximated by:

$$A_\theta^{ext} = \frac{1}{2}B_0 \left[ r - \frac{L}{\pi} \frac{R-1}{R+1} I_1\left(\frac{2\pi r}{L}\right) \cos\left(\frac{2\pi z}{L}\right) \right], \quad (3.4)$$

where  $L$  is the distance between magnetic mirrors (located at  $\pm L/2$ );  $I_1$  is the modified Bessel function of the first kind of order one;  $R$  is the mirror ratio defined by

$$R = \frac{B_{max}}{B_{min}} = \frac{B_z^{ext}(r=0, z=\pm L/2)}{B_z^{ext}(r=0, z=0)}. \quad (3.5)$$

Following the derivation in [?], one can write the envelope function  $\Psi_e(r, z)$  defined by

$$\frac{1}{2m_e} \Psi_e(r, z) = H_0 + e\Phi_0(r, z) - \frac{1}{2m_e} \left[ \frac{P_0}{r} + eA_\theta^{ext}(r, z) \right]^2, \quad (3.6)$$



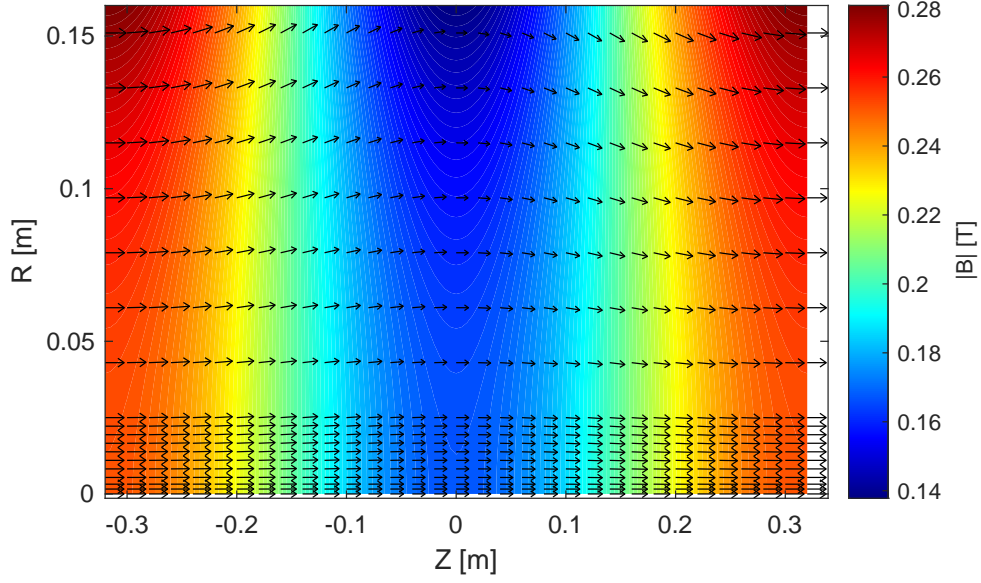


Figure 3.1: Contour of the external magnetic field  $B_0$ . The colour code gives the magnetic field amplitude while the arrows give its direction.

which allows to rewrite the distribution function as

$$f_e^0(r, z, \mathbf{p}) = \frac{\hat{n}_e \hat{r}_b^-}{\pi r} \delta[p_r^2 + p_z^2 - \Psi_e(r, z)] \delta\left[p_\theta - \left(\frac{P_0}{r} + eA_\theta^{ext}(r, z)\right)\right]. \quad (3.7)$$

Given this distribution function, the average flow velocity in the azimuthal direction is

$$V_\theta^0(r, z) = \frac{\int d^3p v_\theta f_e^0}{\int d^3p f_e^0} = \frac{P_0}{m_e r} + \frac{e}{m_e} A_\theta^0(r, z) \quad (3.8)$$

and in the axial direction

$$V_z^0(r, z) = \frac{\int d^3p v_z f_e^0}{\int d^3p f_e^0} = 0. \quad (3.9)$$

Using the same expression for the distribution function, the electron density profile can be obtained by integrating the distribution function over the velocity space:

$$\begin{aligned} n_e^0(r, z) &= \int d^3p f_e^0(r, z, (p)) = \hat{n}_e \frac{\hat{r}_b^-}{r} \int_0^\infty dp_\perp^2 \delta[p_\perp^2 - \Psi(r, z)] \\ &= \hat{n}_e \frac{\hat{r}_b^-}{r} U[\Psi_e(r, z)] \end{aligned} \quad (3.10)$$

where  $U$  is the Heaviside step function. According to this relation, the electron cloud boundaries in  $z$  and  $r$  are therefore determined through the zeroes of  $\Psi_e$

$$\Psi_e^0(r, z) = 0. \quad (3.11)$$

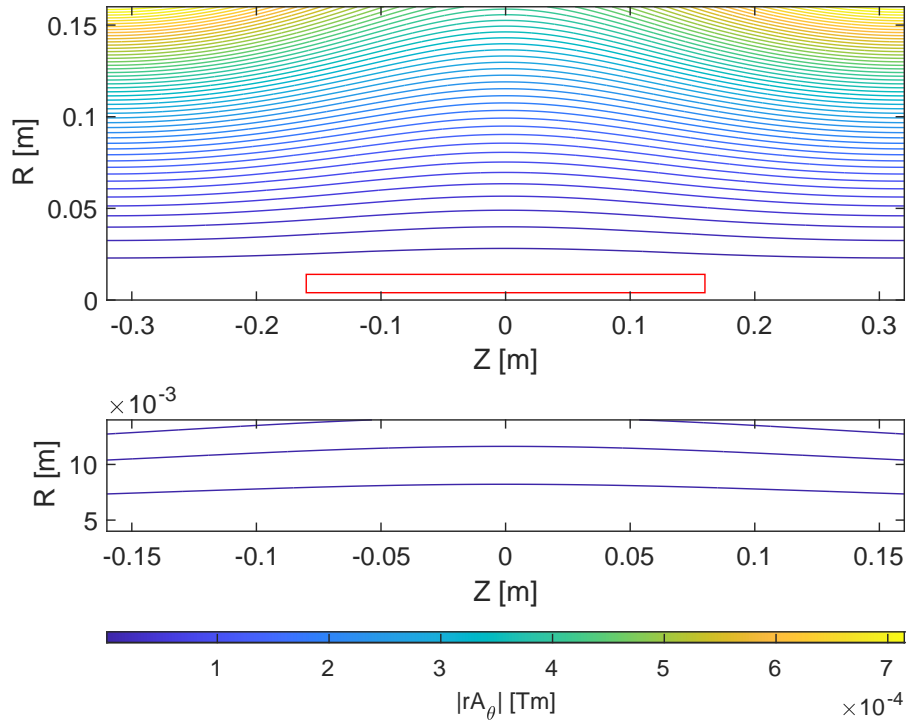


Figure 3.2: Representation of the system magnetic field lines. The top figure show the full simulated space while the bottom figure show a zoom on the region where the electron cloud is present. The red rectangle on the top plot represent the region of zoom.

This equation combined with Poisson's

$$\left[ \frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2} \right] \Phi_0(r, z) = 4\pi e \hat{n}_e \frac{\hat{r}_b^-}{r} U[\Psi_e^0(r, z)], \quad (3.12)$$

and appropriate boundary conditions, form a closed system of non-linear partial differential equations whose solution gives the electron cloud boundaries.

To find an approximation of the solution to this system, one can treat  $e\Phi_0(r, z)/H_0$  as a small parameter as has been done in [?] and write  $\Psi_e(r, z) = \Psi_e^{(0)}(r, z) + \Psi_e^{(1)}(r, z) = 0$  with  $\Psi_e^{(1)}(r, z) = 2m_e e\Phi_0(r, z)$  and

$$\frac{1}{2m_e} \Psi_e^{(0)}(r, z) = H_0 - \frac{1}{2m_e} \left[ \frac{P_0}{r} + \frac{e}{c} A_\theta^{ext}(r, z) \right]^2. \quad (3.13)$$

By approximating  $I_1(2\pi r/L) \simeq \pi r/L$  for  $r \ll L$ , equation 3.13 combined with equation 3.4 give

$$\sqrt{\frac{2H_0}{m_e \omega_{ce}^2}} = \frac{P_0}{m_e \omega_{ce} r_b} + \frac{1}{2} r_b \left[ 1 - \frac{R-1}{R+1} \cos\left(\frac{2\pi z}{L}\right) \right] \quad (3.14)$$

The two solutions of this equation for  $r_b(z)$  determine the inner and outer boundaries of the electron layer at the  $0^{th}$  order. These solutions are

$$r_b(z) = r_b^\pm(z) \equiv r_0 \frac{1 \pm \sqrt{1 - \frac{\omega_{ce} P_0}{H_0} \left[ 1 - \frac{R-1}{R+1} \cos\left(\frac{2\pi z}{L}\right) \right]}}{1 - \frac{R-1}{R+1} \cos\left(\frac{2\pi z}{L}\right)}. \quad (3.15)$$

Where  $\omega_{ce} = \frac{qB_0}{m_e}$  is the electron cyclotron frequency and  $r_0$  is defined as  $r_0 \equiv \sqrt{2H_0/m_e \omega_{ce}^2}$ .

### 3.1 Particle loading

Variable	Value
$\hat{n}_e$	$5 \cdot 10^{14} \text{ m}^{-3}$
$B_0$	$0.21 \text{ T}$
$R$	$1.5$
$P_0$	$3.2 \cdot 10^{-14} \text{ J}$
$H_0$	$8.66 \cdot 10^{-25} \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
$\omega_{pe}$	$\sim 1.2 \cdot 10^9 \text{ s}^{-1}$
$\omega_{ce}$	$\sim 36.9 \cdot 10^9 \text{ s}^{-1}$
$f_z$	$\sim \frac{\langle  v_z  \rangle}{L} \simeq 0.16 \cdot 10^9 \text{ s}^{-1}$
$L$	$0.64 \text{ m}$
$r_a$	$0.0 \text{ m}$
$r_c$	$0.16 \text{ m}$
$l_z$	$0.64 \text{ m}$

Table 3.1: Table of the typical physical quantities used in the simulations for validating FENNECS.

Variable	Value	Normalised value
$dt$	$8 \cdot 10^{-13} \text{ s}$	0.001
$n_z$	128	
$n_r$	[10; 30; 30]	
$r_{lim}$	[0; 4.0; 17.0; 160.0] $mm$	[0; 0.017; 0.072; 0.673]
$dz$	5.0 $mm$	0.021
$dr$	[0.4; 0.4; 4.8] $mm$	[0.0017; 0.0017; 0.020]
$n_{plasma}$	264600	
$n_{blocks}$	1000	

Table 3.2: Typical numerical values used to validate FENNECS.

The implementation of the particle loading according to the distribution function 3.1 is done in several step. First the electron layer boundary is approximated by a piecewise constant function using equation 3.15 where the number of segments and the physical quantities ( $H_0$ ,  $P_0$ ,  $R$ ,  $L$ ,  $B_0$ ) are defined as input parameters. Then, integrating the electron density profile 3.10 in each of the electron cloud slices, the local and global number of electrons is calculated. Given these numbers, the local number of electron is distributed uniformly in the  $z$  direction and according to equation 3.10 in the  $r$  direction for each slice. The result of this particle loading using the physical and numerical parameters described in tables 3.1 and 3.2 is shown for the entire simulation space in figure 3.3. A close-up of this electron-cloud is found in figure 3.4. To illustrate the meaning of slices and piecewise constant particle loading, the initial particle density for 8 slices is represented in figure 3.5.

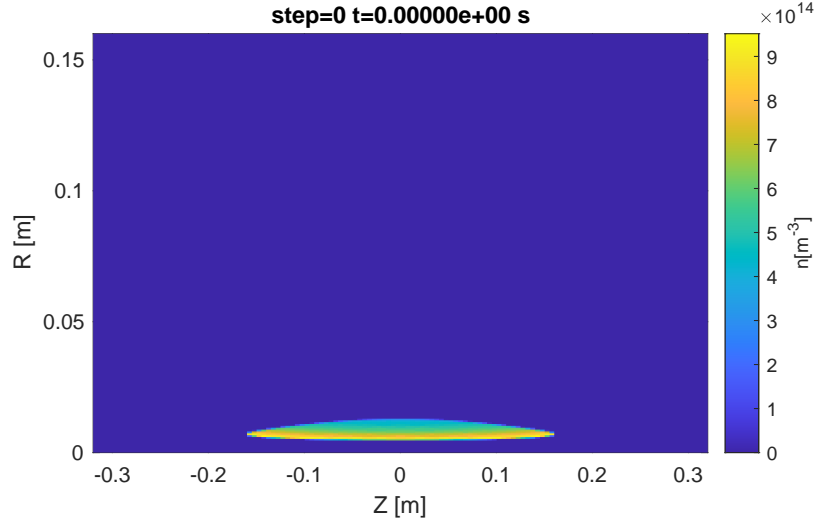


Figure 3.3: Initial particle loading in  $(r, z)$  according to equations 3.15 and 3.10.

Once the position of each macro-electron has been defined, the self-consistent electric potential is computed (figure 3.6). The azimuthal velocity is then calculated for each macro-electron  $k$  according to its position using

$$v_{\theta,k} = \frac{1}{m_e} \left[ \frac{P_0}{r_k} + eA_{\theta}^{ext}(r_k, z_k) \right]. \quad (3.16)$$

Given  $\Phi_0(r, z)$ , the velocity perpendicular to  $v_{\theta}$  is calculated with

$$v_{\perp,k}^2 = \frac{2}{m_e} [H_0 + e\Phi_0(r_k, z_k)] - v_{\theta,k}^2. \quad (3.17)$$

Finally, the radial and longitudinal velocities are  $v_{r,k} = v_{\perp,k} \cos(\psi)$  and  $v_{z,k} = v_{\perp,k} \sin(\psi)$  with  $\psi$  a random variable distributed uniformly over all particles. The initial velocity distribution function evaluated at several positions of the electron cloud can be found in section 3.2 All the random variables are obtained using a Hammersley sequence to limit the numerical noise.

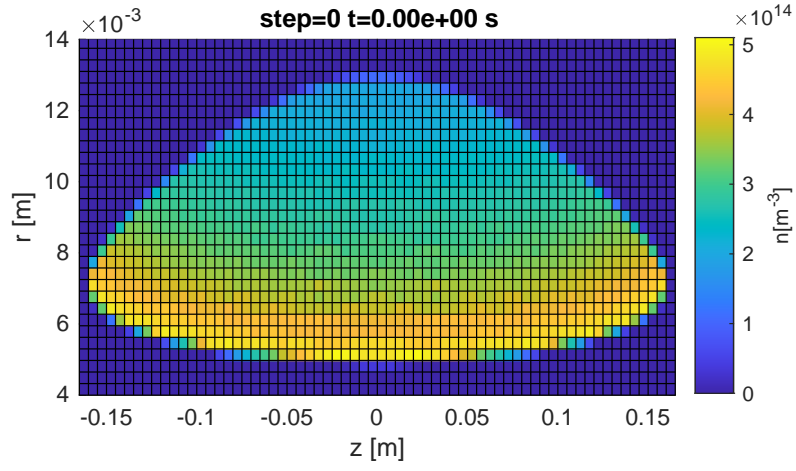


Figure 3.4: Zoom on the electron cloud of figure 3.3 including the FEM grid on which the electrostatic potential is resolved.

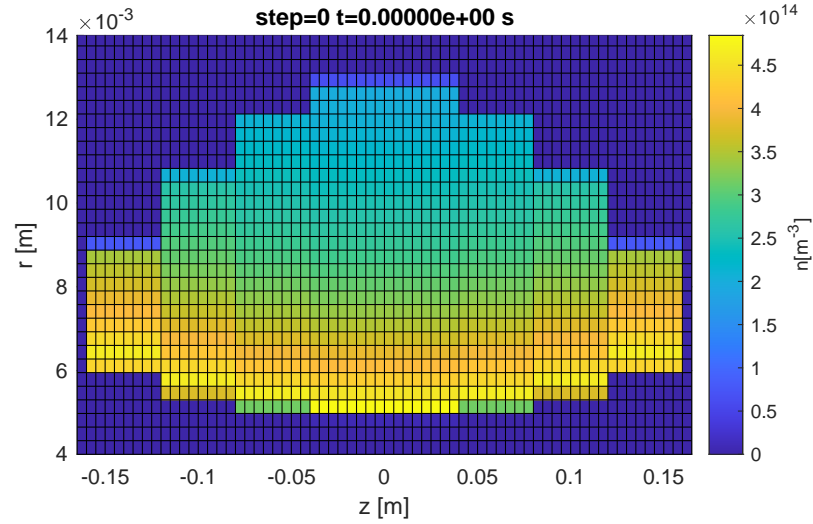


Figure 3.5: Zoom on the electron cloud for an initial electron cloud loaded using only 8 slices. Each step on the cloud boundary delimit a new slice.

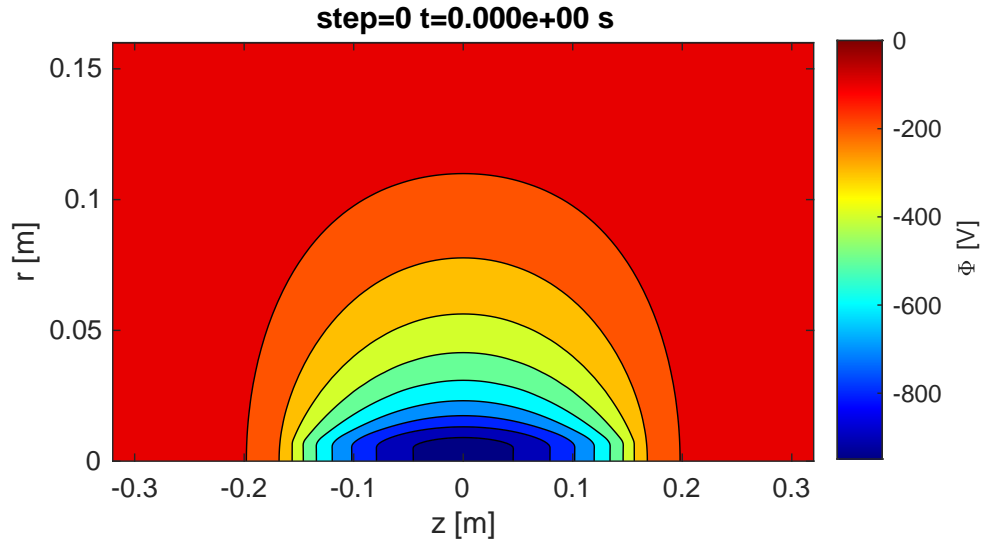


Figure 3.6: Contour of the self consistent electric potential at the beginning of the simulation

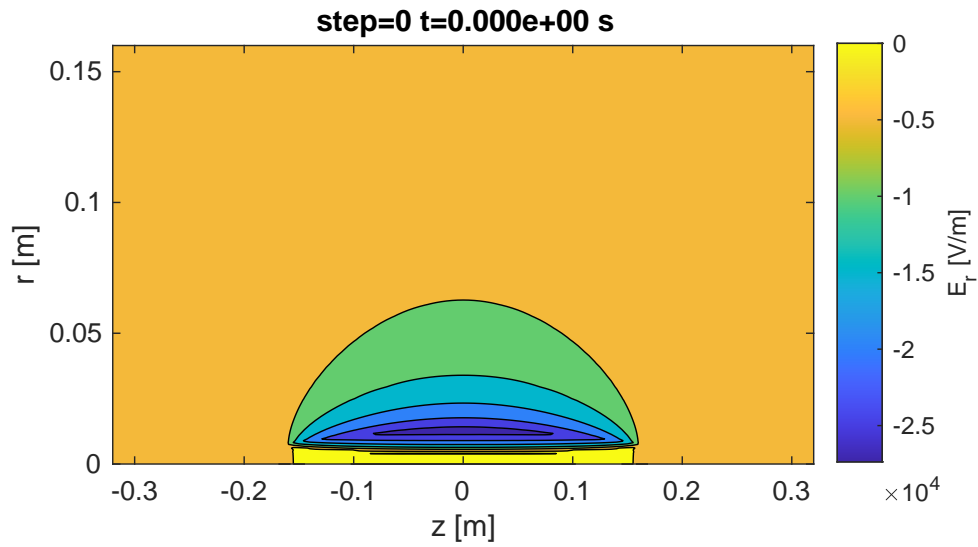


Figure 3.7: Contour of the self consistent electric field in the  $r$  direction at the beginning of the simulation

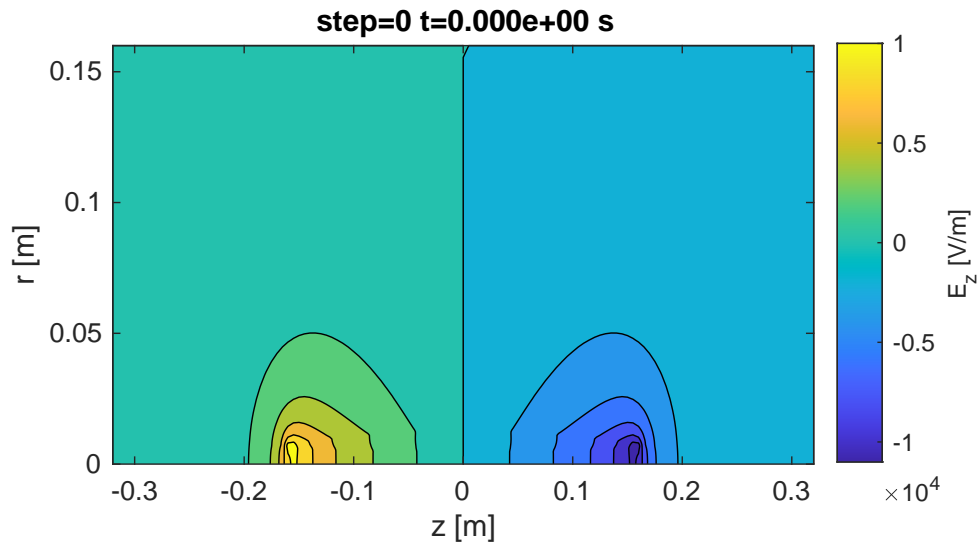


Figure 3.8: Contour of the self consistent electric field in the  $z$  direction at the beginning of the simulation



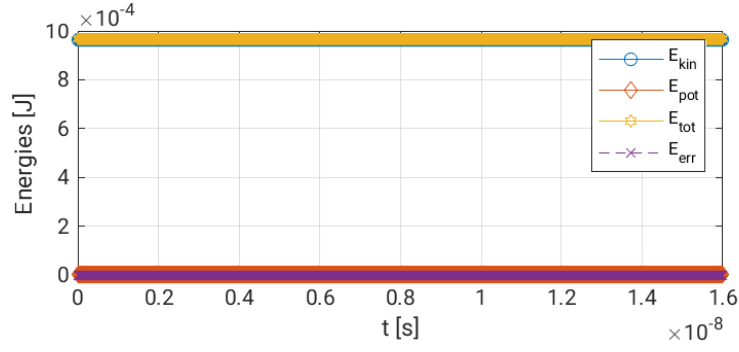
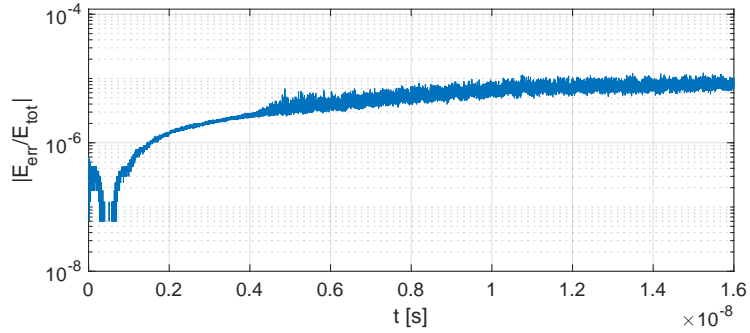


Figure 3.9: Time evolution of the energies during the simulation.

Figure 3.10: Evolution of the relative error on the total energy with  $E_{err} = E_{tot}(t) - E_{tot}(t = 0)$ .

## 3.2 Physical quantities at the end of the simulation

Once simulations have been performed using the initial particle loading described in the precedent section, several simulated quantities can be extracted and plotted to verify if these quantities were conserved.

### Energy conservation

The first quantities to be studied are the kinetic ( $E_{kin}$ ), potential ( $E_{pot}$ ) and total ( $E_{tot} = E_{kin} + E_{pot}$ ) energies summed over all particles. A plot of the time evolution of these quantities and of the error on the total energy conservation  $E_{err} = E_{tot}(t) - E_{tot}(t = 0)$  are shown in figure 3.9. In figure 3.10 the relative error is plotted as a function of time. These graphs clearly show that the energy is well conserved in these simulations.

## 2D quantities

Some 2D quantities can also be represented to verify if the code has conserved the stability of the distribution function. The first quantity is the density shown in figure 3.11 with the red line showing the boundaries of the cloud at initial time. This clearly shows that the cloud remained confined. A contour plot of the electric potential obtained from this density is also shown in figure 3.13. As a further indicator, a figure containing the density in the  $r$  direction for  $z = 0$  at initial and final states is shown in figure 3.12. On top of this graph is also plotted the envelope function  $\Psi(r, 0)$  scaled to better see the points where  $\Psi$  changes its sign. The concordance of the electron cloud boundaries and the envelope function zeroes further validates the code.

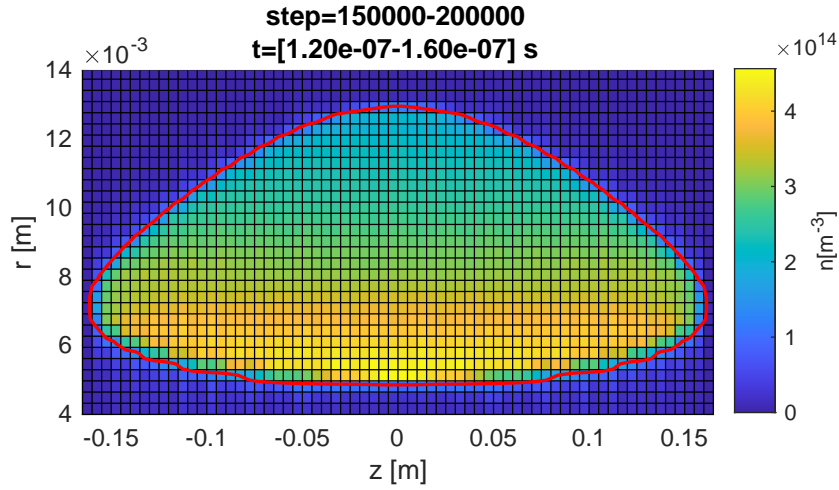


Figure 3.11: Zoom on the electron cloud density at the end of the simulation averaged over 50'000 time steps. The red line shows the contour of the cloud at the beginning of the simulation for comparison.

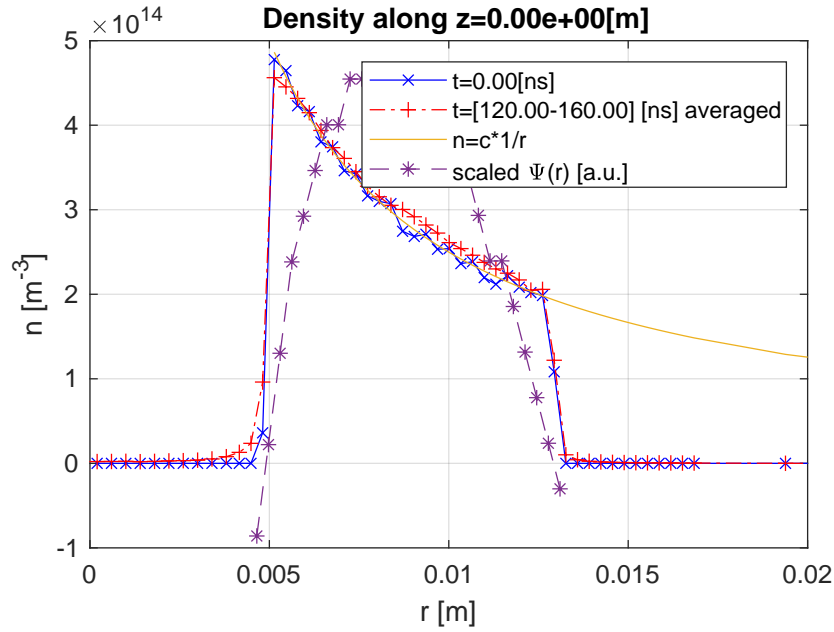


Figure 3.12: Profile of the density along the  $r$  direction taken at  $z = 0$ . The profile represented in blue is the density at the beginning of the simulation and the red profile is the average of the density over 50'000 time steps at the end of the simulation. To verify if the electron cloud boundaries are still defined through the envelope function,  $\Psi(r, z)$  calculated from the simulated electric potential, Hamiltonian  $H$  and canonical angular momentum  $P$  is drawn in purple. We can observe that  $\Psi(r, 0)$  changes its sign at the cloud boundaries.

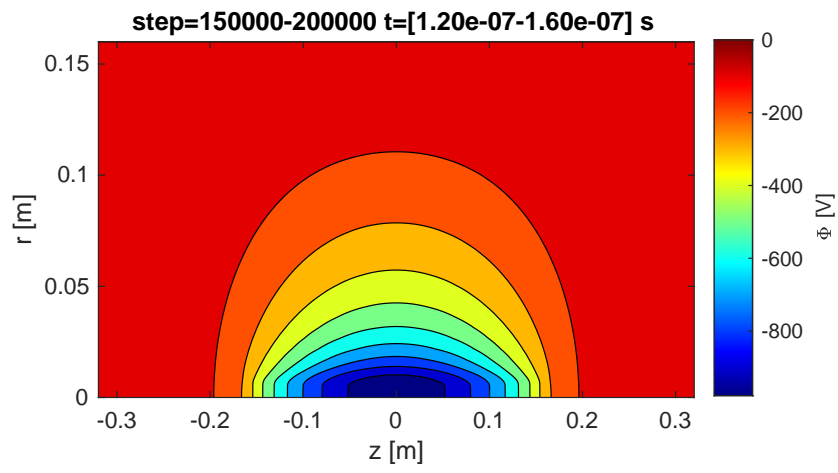


Figure 3.13: Contour of the self consistent electric potential at the end of the simulation averaged over 50'000 time steps.

### Velocity and energy distribution functions

Another indicator used to verify if the distribution function remains the same during the simulation is to plot the particle energy, canonical angular momentum and velocity distribution at several points in space. According to the stable distribution function 3.1 the electron cloud is mono-energetic and the value of the canonical angular momentum is the same for all particles. The distribution of these quantities for all particles has been plotted in figure 3.14 and we can verify that the particles have been loaded with a dirac delta in  $H$  and  $P$  and that little spread was added due to numerical errors. This spread is reduced if the time-step is reduced.

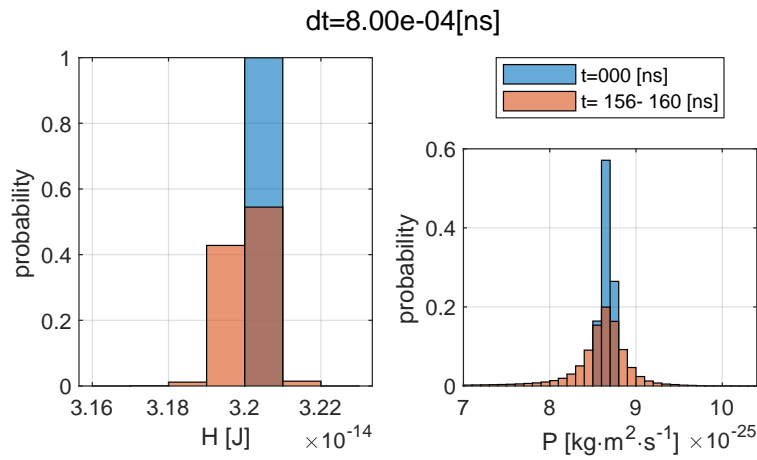


Figure 3.14: Distribution functions of the Hamiltonian  $H$  and the canonical angular momentum  $P$  integrated over  $z$  and  $r$  at the beginning and the end of the simulation.

As with  $H$  and  $P$  the three component velocity distribution has also been represented for the full space (integration over  $(r, z)$ ) in figure 3.15 and at local points in space in figures 3.16, 3.17 and 3.18. This allows a better understanding of the distribution function shape and gives some insight into the changes that occur during the simulation. We can observe that the longitudinal velocity distribution  $V_z$  remains limited in amplitude, while more exchange happens between the radial ( $V_r$ ) and azimuthal ( $V_\theta$ ) velocities. Finally at high radial and axial velocities, the distribution function tends to equilibrate itself between positive and negative values. This can be seen if we look at the axial distribution function in figure 3.18. At the beginning of the simulation more particles have a high positive velocity than high negative velocity, while at the end of the simulation, the number of particles at each velocity is levelled.

dt=8.00e-04[ns] R=[0.00e+00-1.60e-01][m] Z=[-3.20e-01-3.20e-01][m]

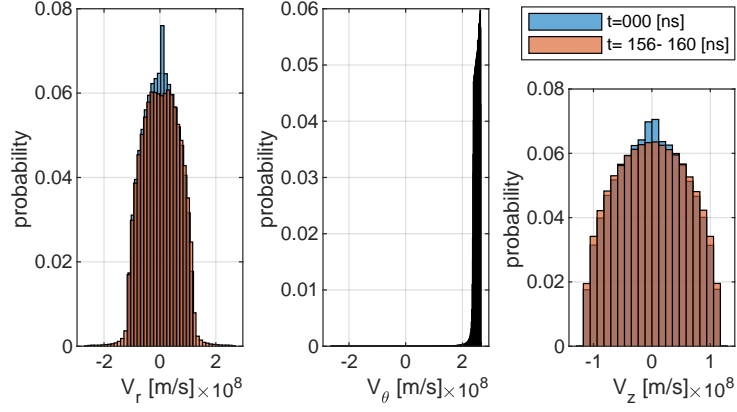


Figure 3.15: Distribution functions of the 3 components of the velocity integrated over  $z$  and  $r$  at the beginning and the end of the simulation.

dt=8.00e-04[ns] R=6.92e-03[m] Z=-1.45e-01[m]

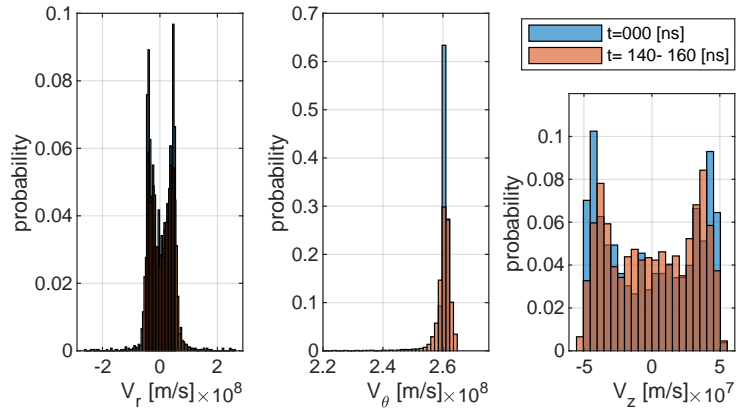


Figure 3.16: Distribution functions of the 3 components of the velocity evaluated in a cell on the left part of the electron cloud at the beginning and the end of the simulation.

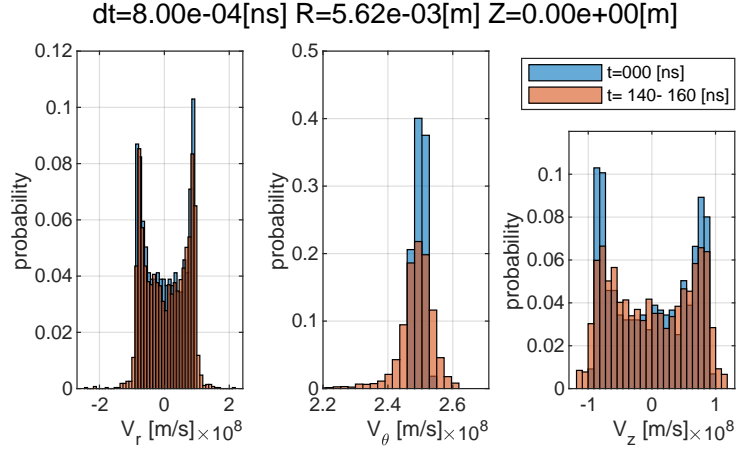


Figure 3.17: Distribution functions of the 3 components of the velocity evaluated in a cell on the bottom center part of the electron cloud at the beginning and the end of the simulation.

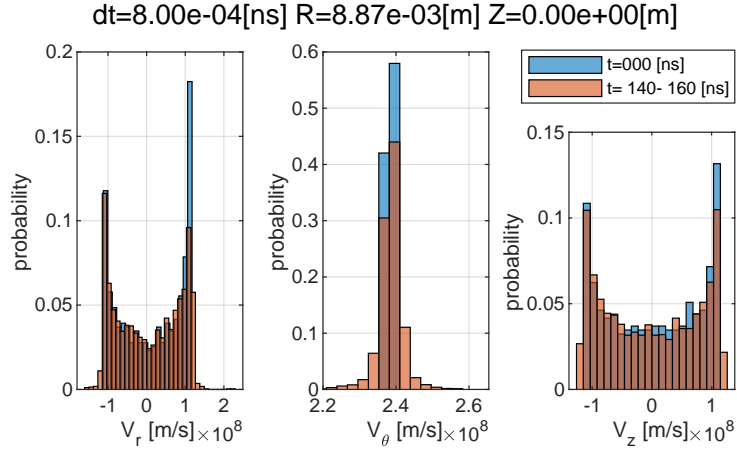


Figure 3.18: Distribution functions of the 3 components of the velocity evaluated in a cell on the top center part of the electron cloud at the beginning and the end of the simulation.

### 3.3 Particles trajectories

To further understand the behaviour of the particles in the simulation, the trace of the particles position has been plotted in both the  $(r, z)$  and  $(x, y)$  planes. These traces are shown in figure 3.19 and 3.20 for particles initialized in two different regions of  $z$ . From these plot, we can observe that the particles follow a cyclotronic motion in the  $(x, y)$  plane coupled with an oscillatory motion in  $z$  due to the magnetic mirror. The peculiarity of these trajectories is that the guiding center is off-center leading to the oscillatory motion observable in the  $r$  direction for the trace projection in the  $(r, z)$  plane.

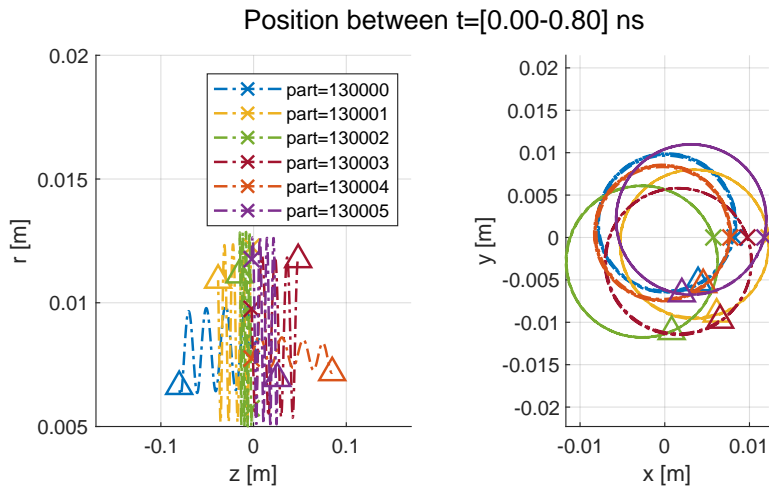


Figure 3.19: Traces of the trajectories of some particles taken in the center region of the electron cloud and projected in the  $(r, z)$  and  $(x, y)$  planes.  $\times$  represent the starting point of the trace and  $\triangle$  its endpoint.

### 3.4 Time evolutions and spectrum

Even if the system is at equilibrium, some fluctuations can still exist in the density or the electric potential. These fluctuations can come from several sources, for example electrons bouncing between the magnetic mirrors, plasma waves appearing due to non-ideal particle loading and discrete effects, particles oscillating in the radial direction due to off-center guiding-center... For these reasons we are interested in the time evolution of the density and electric potential but more specifically in their frequency spectrum. This metric will allow us to determine what are the waves present in this plasma. In this section two point in space have been considered: the first one (figures 3.21 and 3.22) is taken in the center of the electron cloud while the second (figures ?? and ??) is taken close to the left extremity of the electron cloud.



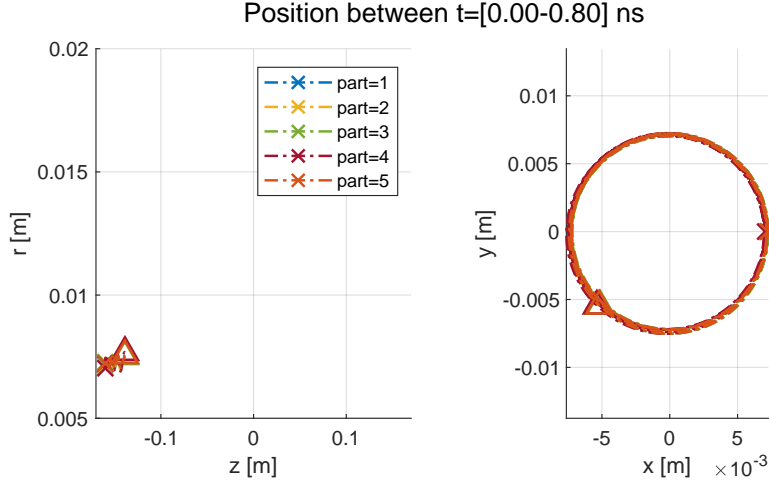


Figure 3.20: Traces of the trajectories of some particles taken in the left region of the electron cloud and projected in the  $(r, z)$  and  $(x, y)$  planes.  $\times$  represent the starting point of the trace and  $\triangle$  its endpoint.

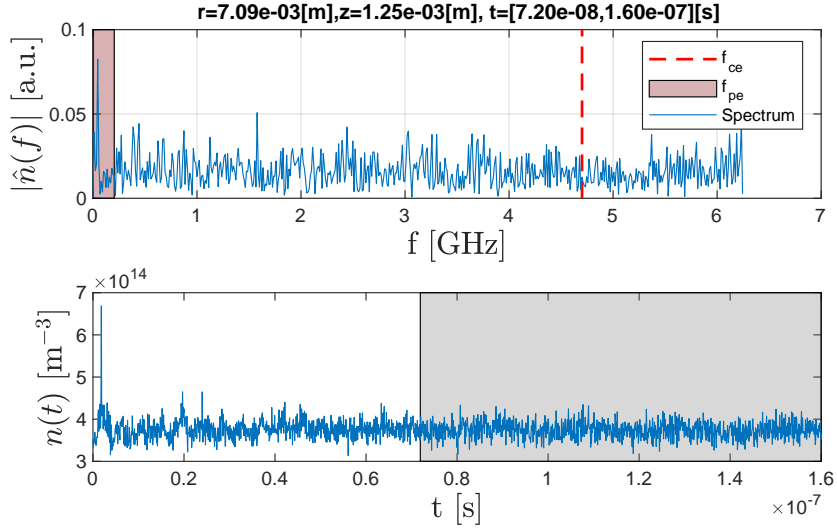


Figure 3.21: Spectrum and time evolution of the density evaluated in the center region of the cloud. The shaded area in the time evolution represent the time period on which the fft has been performed. The red vertical dashed line represent the local cyclotronic frequency and the red shade show the lower and upper limit of the plasma frequency.

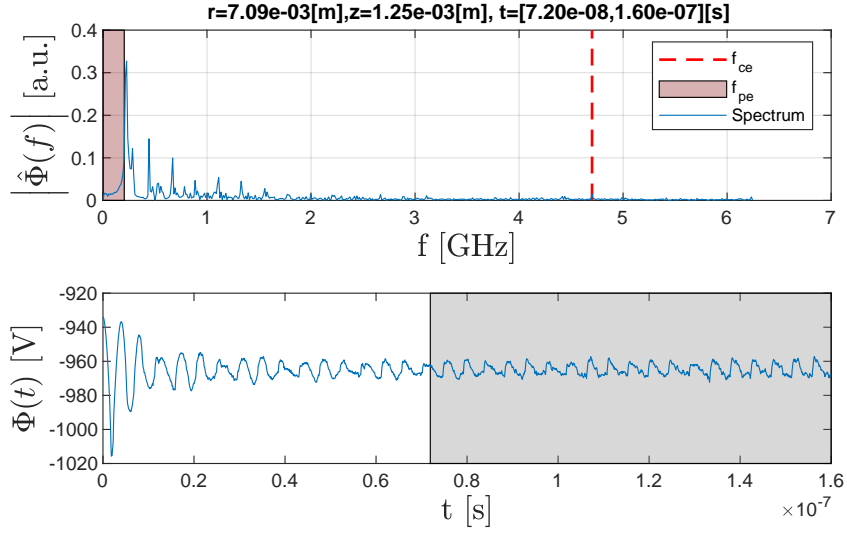


Figure 3.22: Spectrum and time evolution of the electric potential  $\Phi$  evaluated in the center region of the cloud. The shaded area in the time evolution represent the time period on which the fft has been performed. The red vertical dashed line represent the local cyclotronic frequency and the red shade show the lower and upper limit of the plasma frequency.

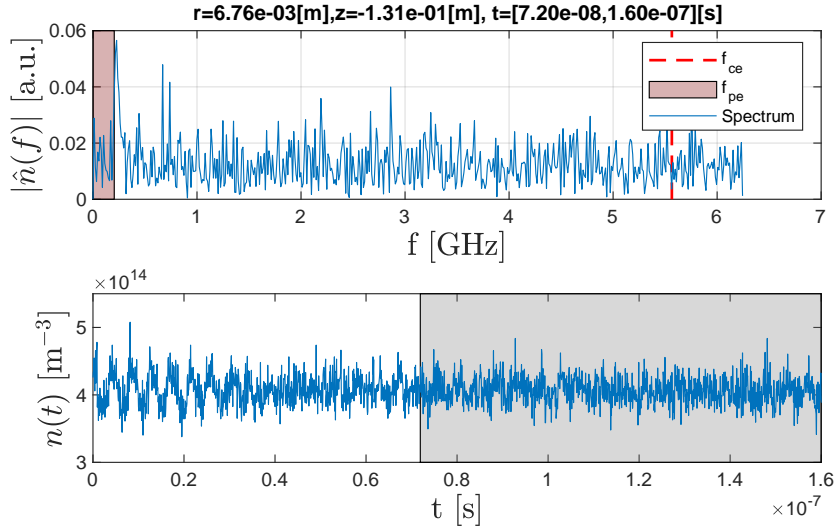


Figure 3.23: Spectrum and time evolution of the density evaluated in the left region of the cloud. The shaded area in the time evolution represent the time period on which the fft has been performed. The red vertical dashed line represent the local cyclotronic frequency and the red shade show the lower and upper limit of the plasma frequency.

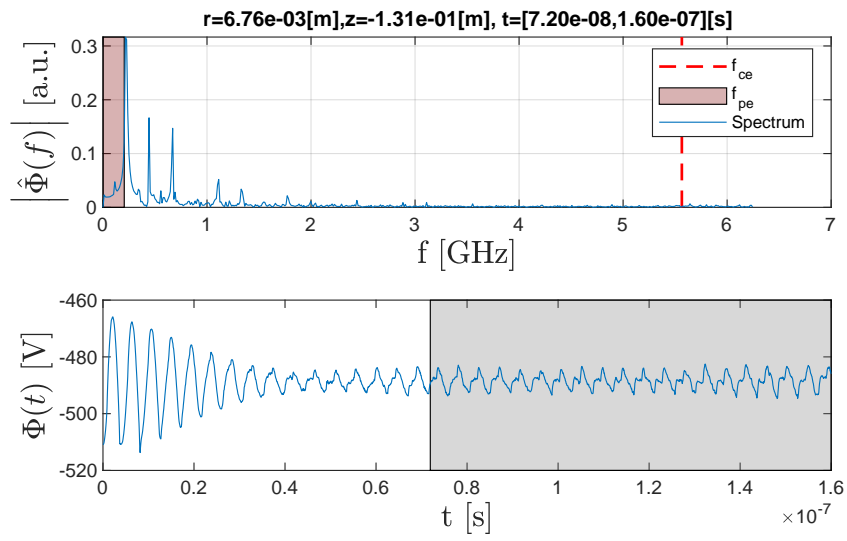


Figure 3.24: Spectrum and time evolution of the electric potential  $\Phi$  evaluated in the left region of the cloud. The shaded area in the time evolution represent the time period on which the fft has been performed. The red vertical dashed line represent the local cyclotronic frequency and the red shade show the lower and upper limit of the plasma frequency.

## Chapter 4

# Finite element method: weighted B-splines

We start from the definition of the Poisson problem with a mix of Dirichlet and Neumann boundary conditions:

$$-\nabla^2 \phi = f \quad (4.1)$$

For a given multivariate function  $\phi$  defined on the volume  $D$ . The boundaries of  $D$  are defined on two regions  $\Gamma_D$  and  $\Gamma_N$  where, respectively, inhomogeneous Dirichlet and natural Neumann boundary conditions are imposed:

$$\phi = g \quad \text{on} \quad \Gamma_D, \quad (4.2)$$

$$\partial^\perp \phi = 0 \quad \text{on} \quad \Gamma_N, \quad (4.3)$$

with  $\partial D = \Gamma_D \cup \Gamma_N$ . We can multiply both sides of Poisson equation 4.2 with a test function  $v$  and integrate the equation over the full domain  $D$ . By doing an integration by part, we get the following weak formulation of Poisson's equation:

$$\int_D \nabla \phi \nabla v = \int_D f v + \int_{\Gamma_N} \partial^\perp \phi v + \int_{\Gamma_D} \partial^\perp \phi v. \quad (4.4)$$

We then separate the solution  $\phi$  into the solution of the homogeneous problem  $u$  and the extension to the full domain of the inhomogeneous part  $\tilde{g}$  :

$$\phi = u + \tilde{g}, \quad (4.5)$$

such that

$$-\nabla^2(u + \tilde{g}) = f \quad \text{with} \quad u = 0 \quad \text{on} \quad \Gamma_D \quad \text{and} \quad \tilde{g} = g \quad \text{on} \quad \Gamma_D, \quad (4.6)$$

If we then impose that  $v = 0$  on  $\Gamma_D$ , we get:

$$\int_D \nabla u \nabla v = \int_D f v - \nabla \tilde{g} \nabla v. \quad (4.7)$$

This equation is then true for any  $v$  and for the case at hand we will define  $v$  using a multivariate b-splines basis. Furthermore we can define the bilinear form

$$a(u, v) = \int_D \nabla u \nabla v, \quad (4.8)$$

and the linear functional

$$\lambda(v) = \int_D f v - \nabla \tilde{g} \nabla v. \quad (4.9)$$

By solving the equation

$$a(u_h, B_i) = \lambda(B_i) \quad (4.10)$$

for the b-splines  $B_i$  of the basis gives us an approximation of  $u(\vec{x}) \simeq u_h(\vec{x})$

$$u_h(\vec{x}) = \sum_{k \in K} u_k b_k(\vec{x}) + \sum_{l \in K_{\Gamma_D}} g_l b_l(\vec{x}), \quad (4.11)$$

where  $K$  is the set of indices of nonzero b-splines  $b_k$  on the domain  $D$ ,  $K_{\Gamma_D}$  the set of indices of nonzero b-splines  $b_k$  on the boundary  $\Gamma_D$  and  $u_k$  are scalar coefficients. Now if by multiplying the b-splines by a weight  $w(\vec{x})$ , they vanish on the boundaries  $\Gamma_D$ , any function represented in this weighted basis will automatically respect the homogeneous Dirichlet boundary conditions. We will then have a solution of the form

$$u(\vec{x}) \simeq u_h(\vec{x}) = \sum_{k \in K} \tilde{u}_k w(\vec{x}) b_k(\vec{x}) + \sum_{l \in K_{\Gamma_D}} g_l b_l(\vec{x}) \quad (4.12)$$

With this definition we can solve the Poisson problem for an approximation  $u_h$  of  $u$  by solving a linear system of equations on this weighted b-spline basis.

#### 4.0.1 Univariate and multi-variate b-splines

A univariate b-spline  $b_h^n$  is defined as a polynomial of degree  $n$  on a bounded support with a grid width  $h$ . It can be constructed recursively and permits the creation of a spline basis with  $n+1$  b-splines capable of representing exactly any polynomial of degree  $\leq n$  on the basis support. For a better visualization of a b-spline basis, a basis of degree 2 with 10 knots is represented in figure 4.1. To construct multivariate b-splines, no unique generalization of univariate b-splines exist. However, a common practice is to create multivariate b-splines by doing a tensor product of univariate b-splines defined in each dimensions. The  $m$ -variate b-spline  $b_{k,h}^n$  of degree  $n_\nu$  is defined as:

$$b_{k,h}^n(x) = \prod_{\nu=1}^m b_{k_\nu, h_\nu}^{n_\nu}(x_\nu) \quad (4.13)$$

A 2d spline of degree 2 is then representend in figure 4.2

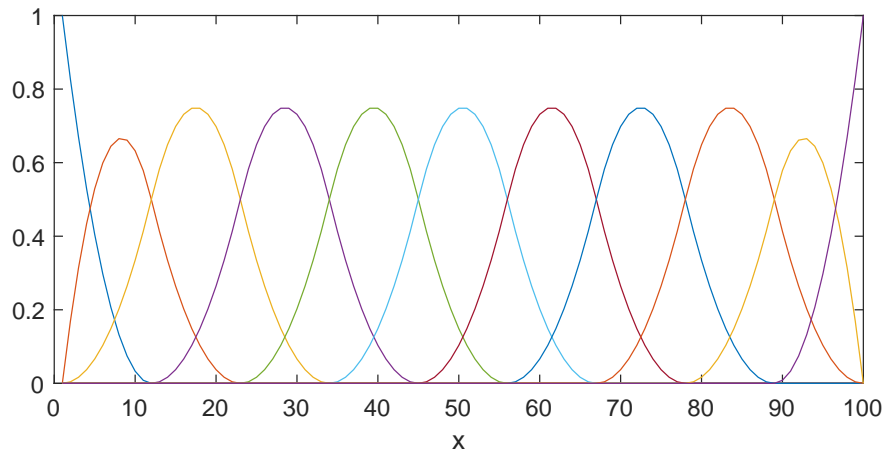


Figure 4.1: Basis of univariate non-periodic b-splines of degree 2 on the interval  $[0, 100]$

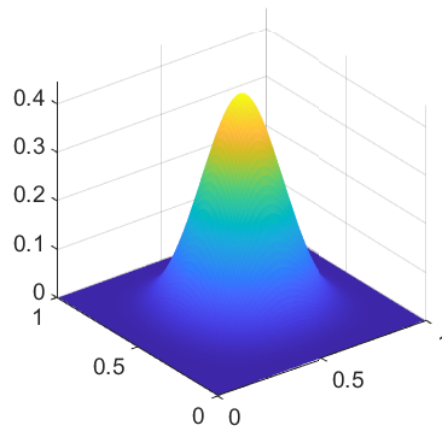


Figure 4.2: Representation of a 2d spline of degree 2 created by the multiplication of two univariate b-splines defined on each dimensions.

### 4.0.2 Representation of the inhomogeneous Dirichlet boundary conditions (transfinite interpolation)

If we represent the Dirichlet boundaries by a combination of weight functions  $w_i$  such that  $w_i = 0$  on  $\Gamma_{D_i}$  and  $\phi = g_i$  on  $\Gamma_{D_i}$ . Rvachev describes a transfinite interpolation method allowing an analytical description of  $\tilde{g}$

$$\tilde{g} = \frac{\sum_i g_i \prod_{k \neq i} w_k}{\sum_i \prod_{k \neq i} w_k} \quad (4.14)$$

In our case we can approximate the external cylinder as an ellipse of center  $(r_0, z_0)$  and radii  $a$  and  $b$  fixed at a potential  $\phi_1$ . The weight function for this boundary is

$$w_1 = \left( \frac{r - r_0}{a} \right)^2 + \left( \frac{z - z_0}{b} \right)^2 - 1. \quad (4.15)$$

The coaxial insert is still represented as a cylinder of constant radius  $r_a$  fixed at potential  $\phi_2$ . For this boundary, the weight is:

$$w_2 = r - r_a. \quad (4.16)$$

Finally in this case the inhomogeneous Dirichlet boundary conditions are imposed with the function

$$\tilde{g} = \frac{\phi_1 w_2 + \phi_2 w_1}{w_1 + w_2} \quad (4.17)$$

### 4.0.3 More complex test case

We manufacture a test case:

$$\phi = \sin(\pi z / L_z) \cos(2\pi r / L_r) \quad (4.18)$$

With  $l_z = 0.48m$  and  $L_r = 0.08m$  in the numeric bspline domain  $z \in [-0.24; 0.24]m$ ,  $r \in [0.001; 0.16]m$ . This analytical function is represented in figure The boundaries are defined at  $r_{low} = 0.003m$  as a straight line, and the top boundary is set as an ellipse of center  $r_0 = 0.155$ ,  $z_0 = 0$ , demi axis in the radial direction  $r_r = 0.02$  and demi axis in the axial direction  $r_z = 0.26$ . Writing Poisson equation for  $\phi$ :  $-\nabla^2 \phi = f$  we obtain the following analytical right hand side:

$$f = 2\pi \sin(\pi z / L_z) \left[ 2\pi \cos(2\pi r / L_r) (L_z^{-2} + L_r^{-2}) + (r L_r)^{-1} \sin(2\pi r / L_r) \right]. \quad (4.19)$$

To obtain the correct Dirichlet boundary conditions, we define the function  $\tilde{g}$  as:

$$\tilde{g} = \sin(\pi z / L_z) \cos(2\pi r / L_r). \quad (4.20)$$

We can then solve the system numerically.

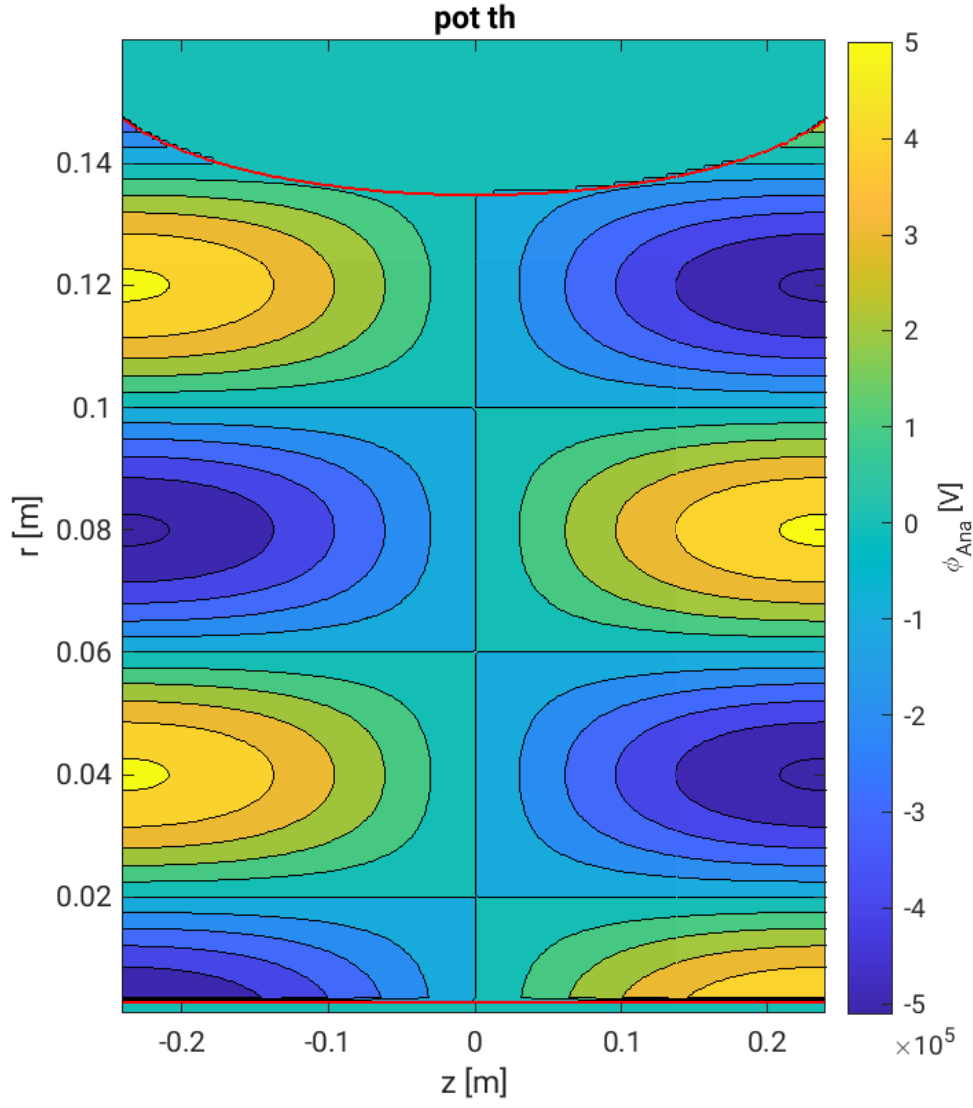


Figure 4.3: Analytical solution of the manufactured more complex test case.



# Chapter 5

## Parallelisation

Due to the relatively high number of simulated particles, and to the good potential for parallelisation of FENNECS it was decided to divide the computation task on multiple machines using MPI (Message Passing interface) libraries. The choice of MPI was made because the cluster we have access to at SPC is composed of machines having only 4 cores which would mean only a small speed-up achievable if pure OpenMP had been used. Furthermore, even if, in the future, the code is run on a big cluster, the code would still be able to scale correctly.

### 5.1 Space decomposition

The easiest way of dividing the computation workload on the different MPI processes would be to divide equally the number of macro-particles between the processes at the beginning of the simulation and let each particle evolve while remaining attached to the same worker. As we plan to implement electron-electron and electron-neutral collisions in the near future, this approach would cause more overhead because it would induce non-locality of the particles simulated by each MPI worker. This is because the simulated particles of a given process would be able to be at any position in the simulated space. Furthermore, as MPI is a library that uses memory distribution, process X is not able to access directly process Y data. The implementation of collisions, using the "naive" parallel implementation, would therefore impose costly communications at each time step.

To alleviate this problem, each worker is assigned a portion of the simulation space and is then responsible of pushing and computing the charge deposition for the particles in its given region of the simulated space. Due to the symmetries of the problem, the division has been implemented in the z direction with non-uniform separation between the workers. This non-uniformity is chosen to better balance the computational load of each process which comes from the non-uniformity of the distribution function. An example of this domain decomposition is shown on figure

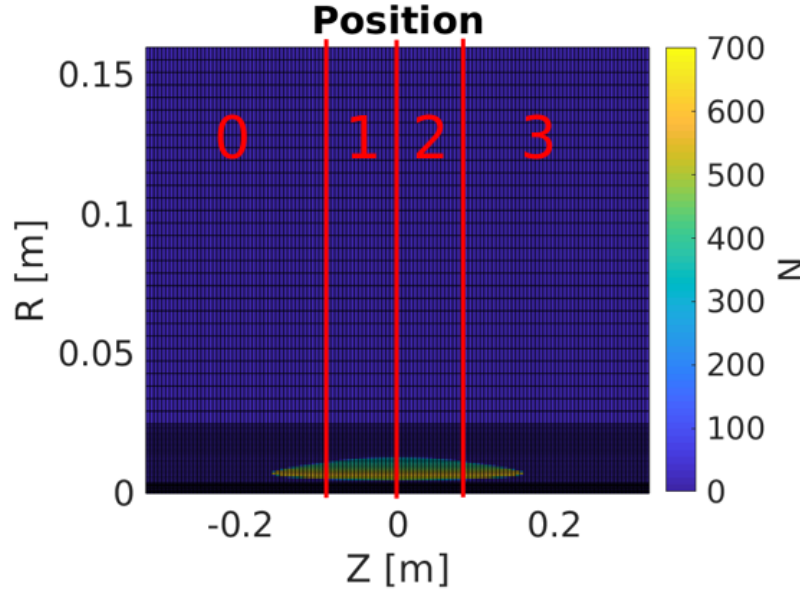


Figure 5.1: Domain decomposition between the MPI processes. The red numbers each represent a given MPI process and the red lines represent the boundaries.

5.1.

## 5.2 Implementation

The MPI parallelisation implied the implementation of several inter-processes communications. The first step was to implement a routine to determine if a particle is leaving a process sub-domain and to which process this particle should be sent. This routine was added to the pre-existing "localisation" routine. Given the list of particles leaving the sub-domain and their destination, two buffers of type particle are filled respectively for the particle moving to the left or right MPI process. These particles are then communicated using non-blocking MPI send of MPI type particle arrays. During these non-blocking send, the incoming particles are received with a non-blocking MPI receive and are included inside the simulation variables.

The second set of communications implemented was for the right hand side (rhs) of Poisson equation. Part of the computation of the rhs is done on each MPI process through the charge deposition of the particles present in the sub-domains. Next, the rhs are collected and summed on the root process through an MPI reduce. Poisson is solved on this process only and the solution is then broadcasted to all other processes.

### 5.3 Measured scaling and improvements

In order to know the ideal scaling of the code, FENNECS was profiled using Intel Inspector on a test case was for a single process. This profiling gave the time spent in each routine, relative to the overall execution time. The results of this profiling are shown in table 5.1. The test case uses the Davidson magnetic mirror stable distribution function described in section 3 with the following numerical parameters:  $N_{step} = 10000$ ,  $dt = 8 \cdot 10^{-12}s$ ,  $nz = 128$ ,  $N_{plasma} = 264600$ ,  $nnr = [10, 30, 30]$ .

routines	total time [s]	time [%]
rhscon	472.8	52.3
getforces	163.6	18.1
comp_velocity	60.2	6.6
localisation	47	5.2
poisson	36.1	4.0
push	35.7	3.9
beam_diagnostics	22.7	2.5

Table 5.1: Profiling of espci2d obtained for the test case used throughout the parallelisation study with  $N = 264600$  macro-particles.

One can observe in table 5.1 that the routines scaling linearly with the number of particles take approximately 90% of the execution time. This information can then be used in Amdahl's law:

$$S = \frac{t_1}{t_p} = \frac{1}{s + (1 - s)/p}, \quad (5.1)$$

to obtain the maximum achievable speed-up for a given number of processes. Here,  $S$  is the speed-up,  $t_1$  and  $t_p$  are the execution times using 1 and  $p$  processes,  $p$  is the number of processes and  $s$  is the fraction of code that does not benefit from an increase of MPI processes.

The actual scaling was first tested using strong scaling which means that the execution time is compared for different number of MPI processes, but with the same problem size ( $N_{plasma} = cst$ ). This scaling and efficiency ( $E = S/p$ ) are represented in figure 5.2.

As can be seen in figure 5.2, there is an important gap between Amdahl's speed-up and the measured speed-up. At first this difference was thought to come from communications overhead. To verify this hypothesis, we estimated the ratio between the number of exchanged particles  $N_{transf}$  and the number of local particles  $N_{local}$  at one process boundary:

$$\frac{N_{transf}}{N_{local}} = \frac{\Gamma S_{cyl} dt}{N_{local}} = \frac{n S_{cyl} v_z dt}{n S_{cyl} (L_z/p)} = \frac{v_z p dt}{L_z}, \quad (5.2)$$

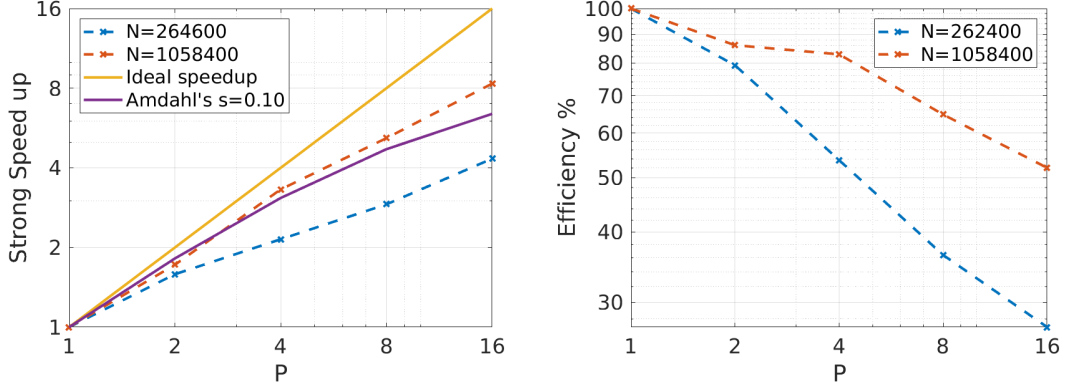


Figure 5.2: Strong speed-up(left) and efficiency(right) of parallel FENNECS measured for 1 to 16 MPI processes. This scaling is measured for two cases, one using  $N_{plasma} = 264600$  macro-particles and one using  $N_{plasma} = 1058400$  macro-particles.

with  $S_{cyl}$  the surface of the cylinder,  $\Gamma$  the particle flux,  $dt$  the time-step,  $v_z$  the longitudinal velocity,  $L_z$  the simulated space  $z$  dimension and  $p$  the number of MPI processes. Given the following numeric values:

$$dt \simeq 10^{-12}s, \quad v_z \simeq 6 \cdot 10^7 m/s, \quad L_z \simeq 0.30m, \quad (5.3)$$

we obtain

$$\frac{N_{transf}}{N_{local}} \simeq p \cdot 2 \cdot 10^{-4} \quad (5.4)$$

For  $N=264600$  macro-particles the average number of exchanged particles at each process boundary is therefore

$$N_{transf} \simeq 40$$

. This small number of exchanged particles and the small size of the grid (70\*128) cannot explain the measured speed-up. The reason of the poor results come from the fact that, due to the non-uniformity of the distribution function, the computation load balancing is poor. For example with 16 MPI processes there is on average a ratio of 40% between the process having the smallest number of particles and the one having the biggest number.

To improve the load balancing, two approaches have been used. The first one is to the space decomposition at arbitrary positions and not at grid points as it was done until then. This becomes important for high number of processes because the number of grid columns assigned to each process becomes small. For the particular case studied in this section simulated with 16 MPI processes, some processes were only responsible of 4 grid columns. The second approach used to improve the load balancing was to implement Open Multi-Processing (OpenMP) parallelisation at the node level. Because OpenMP use shared memory parallelism, many compute units can be assigned to the same domain. This reduces the overall number of domains needed in the decomposition and significantly improves the load balancing in each domain. Furthermore, the amount of communications is reduced.

# Chapter 6

## Azimuthal rotation frequency study

To better understand an electron cloud ring, trapped inside a magnetic mirror, response to an increase of particle density, simulations were performed at several value of  $n_e$ . This quantity will also be represented by the Brillouin ratio  $2\omega_{pe}^2/\omega_{ce}^2 = 2n_em_e/(\epsilon_0 B_0^2)$  in this study.

### 6.1 Davidson stable force balance

In the bulk of the electron cloud, we can use the fluid force balance in the radial direction and neglect the pressure forces. We obtain

$$-m_e\omega_{re}^2 r = -q_e E_r(r) - q_e\omega_{re} r B_z - \frac{\nabla \cdot \mathcal{P}}{n} \cdot \mathbf{e}_r \quad (6.1)$$

Where the magnetic field  $B_z$  is derived from equation 3.4:

$$B_z(r, z) = \frac{1}{r} \frac{\partial}{\partial r} [r A_\theta^{ext}(r, z)] = B_0 \left[ 1 - \frac{R-1}{R+1} I_0\left(\frac{2\pi r}{L}\right) \cos\left(\frac{2\pi z}{L}\right) \right], \quad (6.2)$$

and  $E_r$  is calculated using Gauss's law given the electron density of equation 3.10. This gives for the radial electric field inside the cloud:

$$\frac{1}{r} \frac{\partial}{\partial r} r E_r = \frac{-q_e n_e(r)}{\epsilon_0} = \frac{-q_e \hat{n}_e r_b^-}{\epsilon_0 r} \quad | \quad r_b^- < r < r_b^+ \quad (6.3)$$

$$E_r = -\frac{q_e \hat{n}_e r_b^-}{\epsilon_0} \left( 1 - \frac{r_b^-}{r} \right), \quad (6.4)$$

where  $\hat{n}_e$  is a constant. The pressure profile is defined as:

$$\underline{\mathcal{P}} = n \underline{T} \quad (6.5)$$

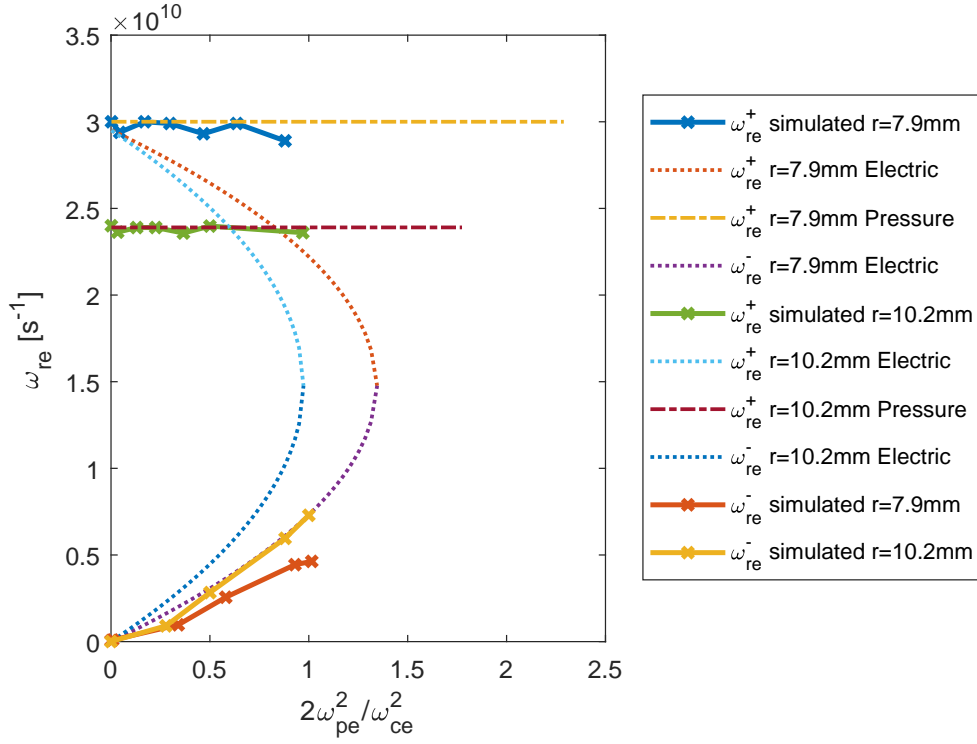


Figure 6.1: Azimuthal rotation frequency for different values of the Brillouin ration

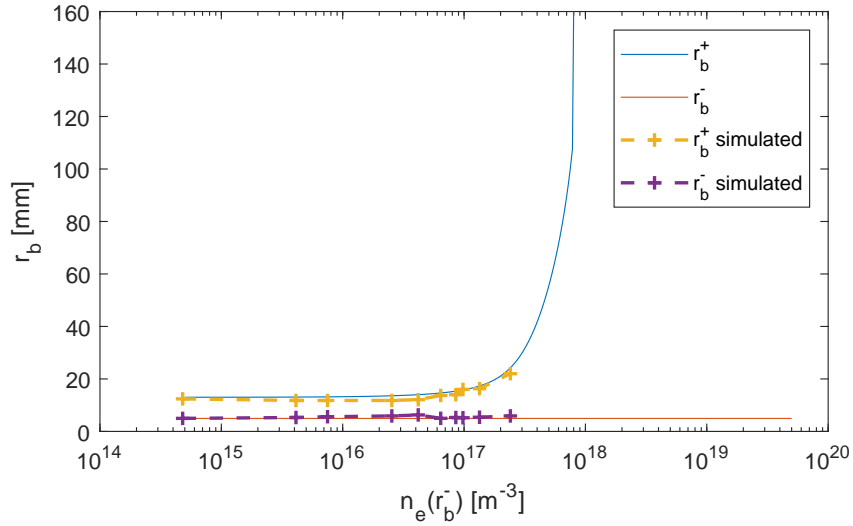


Figure 6.2: Evolution of the radial limits of the electron cloud as a function of the electron density at  $r_b^-$ . These radii are obtained by computing the zeros of the envelope function 3.11.

where  $\underline{T}$  is the temperature tensor. Calculating the distribution function 2nd order moment gives this tensor:

$$\begin{aligned}\underline{T} &= \frac{1}{n} \int \frac{(\mathbf{p} - m_e \mathbf{V})(\mathbf{p} - m_e \mathbf{V})}{2m_e} f_e d^3v \\ &= \left[ H_0 + q_e \Phi(r, z) - \frac{1}{2m_e} \left( \frac{P_0}{r} + q_e A_\theta^{ext}(r, z) \right)^2 \right] (\mathbf{e}_{rr} + \mathbf{e}_{zz}),\end{aligned}\quad (6.6)$$

where  $\mathbf{V}$  is the fluid velocity vector. The pressure force can therefore be calculated as follow:

$$\frac{\nabla \cdot \mathcal{P}}{n} \cdot \mathbf{e}_r = \frac{1}{rn} \frac{\partial}{\partial r} (rn T_{rr}) = \frac{1}{r} \frac{\partial}{\partial r} (r T_{rr}) + \frac{T_{rr}}{n} \frac{\partial}{\partial r} (n) = \frac{\partial T_{rr}}{\partial r} + \frac{T_{rr}}{r} - \frac{T_{rr}}{r} = \frac{\partial T_{rr}}{\partial r} \quad (6.7)$$

With this equation for the temperature, the pressure force can be derived:

$$-\frac{\nabla \cdot \mathcal{P}}{n} \cdot \mathbf{e}_r = -\frac{\partial T_{rr}}{\partial r} = -q_e \frac{\partial \Phi}{\partial r} + \frac{1}{m_e} \left( \frac{P_0}{r} + q_e A_\theta^{ext} \right) \left( \frac{\partial}{\partial r} q_e A_\theta^{ext} - \frac{P_0}{r^2} \right). \quad (6.8)$$

Combining equations 6.1, 6.2, 6.4 and 6.8 we get:

$$\begin{aligned}m_e \omega_{re}^2 r - q_e \omega_{re} r B_0 \left[ 1 - \frac{R-1}{R+1} I_0 \left( \frac{2\pi r}{L} \right) \cos \left( \frac{2\pi z}{L} \right) \right] + \frac{q_e^2 \hat{n}_e}{\epsilon_0} r_b^- \left( 1 - \frac{r_b^-}{r} \right) \dots \\ - \frac{q_e^2 \hat{n}_e}{\epsilon_0} r_b^- \left( 1 - \frac{r_b^-}{r} \right) + \frac{1}{m_e} \left( \frac{P_0}{r} + q_e A_\theta^{ext} \right) \left( \frac{\partial}{\partial r} q_e A_\theta^{ext} - \frac{P_0}{r^2} \right) = 0.\end{aligned}\quad (6.9)$$

This is a second order polynomia in  $\omega_{re}$  which has the two following solutions:

$$\omega_{re}^\pm = \frac{q_e B_z(r, z)}{2m_e} \pm \frac{q_e}{2m_e} \sqrt{B_z^2 - \frac{4}{r q_e^2} \left( \frac{P_0}{r} + q_e A_\theta^{ext} \right) \left( \frac{\partial}{\partial r} q_e A_\theta^{ext} - \frac{P_0}{r^2} \right)}, \quad (6.10)$$

$$\begin{aligned}\omega_{re}^+ &= \frac{q_e A_\theta(r, z)}{r m_e} + \frac{P_0}{r^2 m_e} \\ \omega_{re}^- &= \frac{q_e}{r m_e} \frac{\partial A_\theta(r, z)}{\partial r} + \frac{P_0}{r^2 m_e}\end{aligned}\quad (6.11)$$

If we had neglected the pressure force, we would have obtained:

$$\omega_{re}^\pm = \frac{\omega_{ce}(r, z)}{2} \left( 1 \pm \sqrt{1 - \frac{4m_e \hat{n}_e}{\epsilon_0 B^2(r, z)} \left( \frac{r_b^-}{r} - \left( \frac{r_b^-}{r} \right)^2 \right)} \right) \quad (6.12)$$

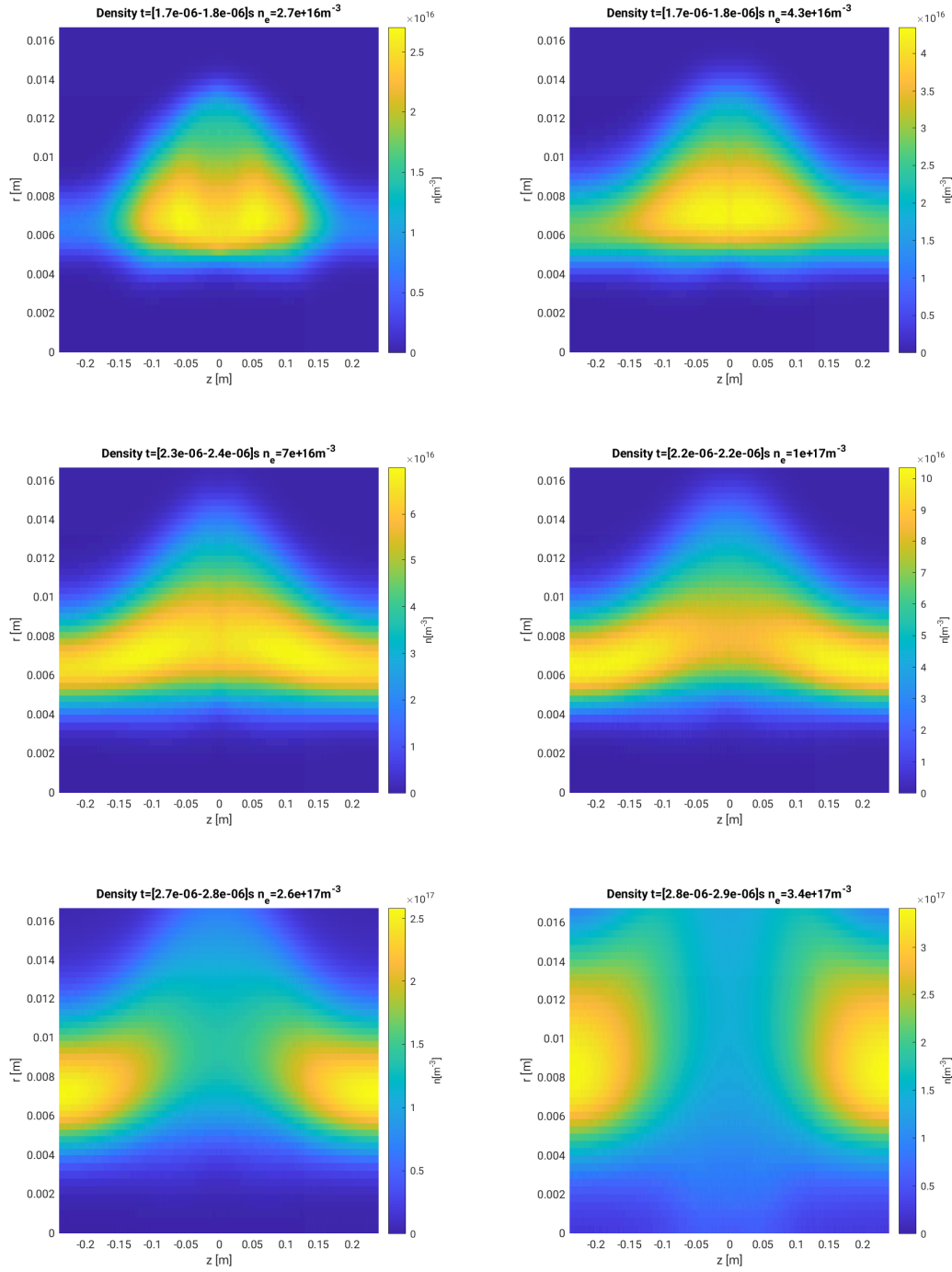


Figure 6.3: Averaged density at the end of simulation for different values of the initial density. The rotation frequency evolution follow the higher branch of figure 6.1.



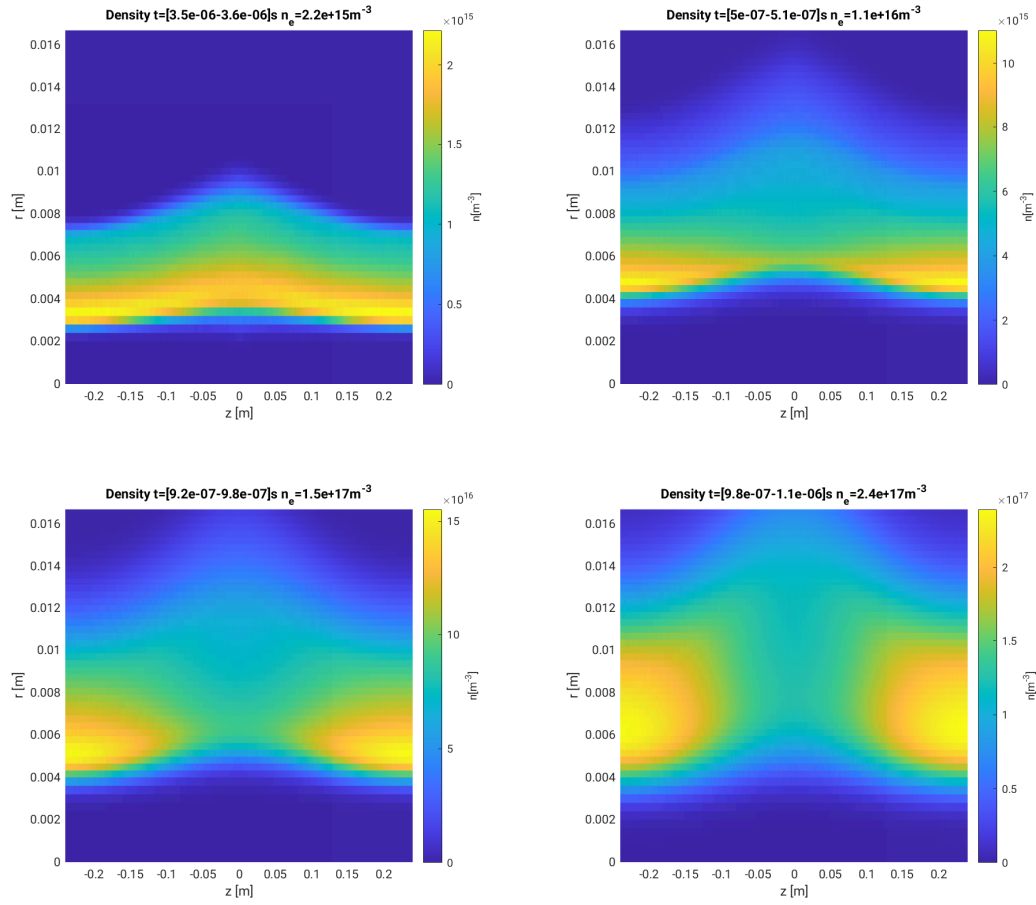


Figure 6.4: Averaged density at the end of simulation for different values of the initial density. The rotation frequency evolution follow the lower branch of figure 6.1.

# Chapter 7

## Characteristic time scales

In a typical gyrotron gun assembly several time scales need to be taken into account. These are:

$$\tau_{pe} = 2\pi \sqrt{\frac{\epsilon_0 m_e}{e^2 n_e}} \quad (7.1)$$

$$\tau_{ce} = 2\pi \frac{\gamma m_e}{eB} \quad (7.2)$$

$$\tau_{ion} = n_p \sigma_{ion} v_e \quad (7.3)$$

$$t_{acc} = \tau_{ion} \ln \frac{n_{e,end}}{n_{e,0}} \quad (7.4)$$

$$\tau_{therm} = n_e \sigma_{therm} v_e \quad (7.5)$$

where  $\tau_{pe}$  is the electron plasma wave period,  $\tau_{ce}$  is the electron cyclotron period,  $\tau_{ion}$  is the characteristic time between two ionization events for an incoming electron;  $t_{acc}$  is the time needed to obtain the density  $n_{e,end}$  in the confinement region if no electron is lost during this time; finally  $\tau_{therm}$  is the electron cloud thermalisation time. In the equations  $n_p$  is the neutral gas density present in the gyrotron;  $v_e$  is the incoming electron velocity. The ionisation cross section  $\sigma_{ion}$  is calculated using the Binary-encounter-dipole model for electron impact ionization [?]. The electron-electron thermalization cross-section is calculated using the plasma Coulomb collision model:

$$\sigma_{therm} = 8\pi \frac{e^4}{(4\pi\epsilon_0)^2 m_e^2 v^4} \ln \left( \frac{\sqrt{\epsilon_0 T_e / e^2 n_e}}{e^2 / 2\pi\epsilon_0 m_e v^2} \right) \quad (7.6)$$

# Chapter 8

## Maximum density estimate

Given a charged particle subjected to an electric field and an arbitrary magnetic field we can write the total energy conservation between position 1 and position 2:

$$\frac{1}{2}m(v_{\perp,1}^2 + v_{\parallel,1}^2) + q\phi_1 = \frac{1}{2}m(v_{\perp,2}^2 + v_{\parallel,2}^2) + q\phi_2, \quad (8.1)$$

where  $\perp$  and  $\parallel$  denotes the component perpendicular and parallel to the magnetic field lines. Under the assumption of adiabaticity, the particle magnetic moment

$$\mu = \frac{mv_{\perp}^2}{2B} \quad (8.2)$$

is conserved along the particle trajectory. We can therefore rewrite equation 9.1 as:

$$\mu B_1 + \frac{1}{2}mv_{\parallel,1}^2 + q\phi_1 = \mu B_2 + \frac{1}{2}mv_{\parallel,2}^2 + q\phi_2. \quad (8.3)$$

For the particle to be lost in the direction parallel to the magnetic field line, we need that at any position along the magnetic field line the parallel kinetic energy stays positive which leads to the following loss condition for a charged particle at position 1:

$$\mu(B_1 - B_2) + \frac{1}{2}mv_{\parallel,1}^2 + q(\phi_1 - \phi_2) \geq 0. \quad (8.4)$$

If we consider a zero parallel kinetic energy at position 1 and that the magnetic field amplitude at 1 is lower than at 2, we can derive a criterion on the particle kinetic energy such that all particles that have a kinetic energy lower than a threshold are lost.

$$E_{kin} \leq -qB_1 \frac{\phi_1 - \phi_2}{B_1 - B_2} \quad (8.5)$$

## 8.1 Case of an annular electron layer in a coaxial geometry with magnetic mirror

In the approximation of an infinite length annular electron layer of constant density

$$n_e(r) = \begin{cases} 0 & a < r < r_- \\ \hat{n}_e & r_- \leq r \leq r_+ \\ 0 & r_+ < r < b \end{cases} \quad (8.6)$$

we can write the electric potential in the geometry as (see appendix B):

$$\phi(r) = \begin{cases} [(\phi_- - \phi_a) \ln(r) + \phi_a \ln(r_-) - \phi_- \ln(a)] \frac{1}{\ln(r_-/a)} & a < r < r_- \\ -\frac{e\hat{n}_e}{4\epsilon_0}(r^2 - r_+^2) + \phi_+ + \left[ \phi_+ - \phi_- + \frac{e\hat{n}_e}{4\epsilon_0}(r_+^2 - r_-^2) \right] \frac{\ln(r/r_+)}{\ln(r_+/r_-)} & r_- \leq r \leq r_+ \\ [(\phi_b - \phi_+) \ln(r) + \phi_+ \ln(b) - \phi_b \ln(r_+)] \frac{1}{\ln(b/r_+)} & r_+ < r < b \end{cases} \quad (8.7)$$

where

$$\begin{aligned} \phi(r_-) = \phi_- = & \left\{ -\frac{e\hat{n}_e}{2\epsilon_0} r_-^2 \ln\left(\frac{r_-}{a}\right) \ln\left(\frac{r_+}{r_-}\right) \right. \\ & \left. + \left[ \frac{e\hat{n}_e}{2\epsilon_0}(r_+^2 - r_-^2) \left( \ln\left(\frac{b}{r_+}\right) + \frac{1}{2} \right) + \phi_b \right] \ln\left(\frac{r_-}{a}\right) + \phi_a \ln\left(\frac{b}{r_-}\right) \right\} \frac{1}{\ln(b/a)} \end{aligned} \quad (8.8)$$

and

$$\begin{aligned} \phi(r_+) = \phi_+ = & \left\{ \frac{e\hat{n}_e}{2\epsilon_0} r_-^2 \ln\left(\frac{b}{r_+}\right) \ln\left(\frac{r_+}{r_-}\right) \right. \\ & \left. + \left[ \frac{e\hat{n}_e}{2\epsilon_0}(r_+^2 - r_-^2) \left( \ln\left(\frac{r_+}{a}\right) - \frac{1}{2} \right) + \phi_a \right] \ln\left(\frac{b}{r_+}\right) + \phi_b \ln\left(\frac{r_+}{a}\right) \right\} \frac{1}{\ln(b/a)} \end{aligned} \quad (8.9)$$

are the electric potential value at the interface between the electron cloud and vacuum, at respectively the radial lower and upper interfaces.

To calculate the radial displacement  $\Delta r$  from position 1 to position 2, we can integrate the trajectory between axial position 1 and 2

$$\Delta r = \int_{z_1}^{z_2} \frac{B_r(r, z')}{B_z(r, z')} dz'. \quad (8.10)$$

Which for the case of Davidson magnetic mirror configuration defined by the vector potential of equation 3.4 can be written as

$$\Delta r = \int_{z_1}^{z_2} \frac{-B_0 \frac{R-1}{R+1} I_1\left(\frac{2\pi r}{L}\right) \sin\left(\frac{2\pi z}{L}\right)}{B_0 \left[ 1 - \frac{R-1}{R+1} I_0\left(\frac{2\pi r}{L}\right) \cos\left(\frac{2\pi z}{L}\right) \right]} dz'. \quad (8.11)$$

If we consider radial positions where  $r \ll L$  we can approximate the axial magnetic field as constant along the axial direction.

$$\begin{aligned} \Delta r & \simeq \int_{z_1}^{z_2} \frac{R-1}{R+1} I_1\left(\frac{2\pi r}{L}\right) \sin\left(\frac{2\pi z}{L}\right) dz' \\ & = \frac{R-1}{R+1} I_1\left(\frac{2\pi r}{L}\right) \left( \cos\left(\frac{2\pi z_2}{L}\right) - \cos\left(\frac{2\pi z_1}{L}\right) \right) \end{aligned} \quad (8.12)$$

## Chapter 9

# Estimated gained energy for particles created with 0 kinetic energy

If we consider the conservation of energy between the creation position (1) and the position of minimum electric potential (2):

$$\frac{1}{2}m(v_{\perp,1}^2 + v_{\parallel,1}^2) + q\phi_1 = \frac{1}{2}m(v_{\perp,2}^2 + v_{\parallel,2}^2) + q\phi_2, \quad (9.1)$$

where  $\perp$  and  $\parallel$  denotes the component perpendicular and parallel to the magnetic field lines. Under the assumption of adiabaticity, the particle magnetic moment

$$\mu = \frac{mv_{\perp}}{2B} \quad (9.2)$$

is conserved along the particle trajectory. We can therefore rewrite equation 9.1 as:

$$\mu B_1 + \frac{1}{2}mv_{\parallel,1}^2 + q\phi_1 = \mu B_2 + \frac{1}{2}mv_{\parallel,2}^2 + q\phi_2. \quad (9.3)$$

Due to 0 kinetic energy at time  $t = 0$ ,  $\mu$  is zero. This means:

$$\frac{1}{2}m(v_{\parallel,2}^2) = q(\phi_2 - \phi_1), \quad (9.4)$$

# Chapter 10

## Diocotron Modes

As solving the Pic code in 3D would be very computationally expensive, we decided to first run the simulations in 2D ( $r, z$ ), then determine the linear stability to diocotron modes by solving an eigen value equation obtained using the fluid-Poisson model.

### 10.1 Case of axial uniformity (infinite length coaxial configuration)

For a TP4 project a first eigenvalue solver for the following diocotron eigenvalue equation [?]:

$$\frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial}{\partial r} \delta\phi^l(r) - \frac{l^2}{r^2} \delta\phi^l(r) = - \frac{l}{\omega_{ce} r \omega - l \omega_{re}^-(r)} \frac{\partial}{\partial r} \omega_{pe}^2(r) \quad (10.1)$$

has been implemented. This equation is solved using finite differences in space and an eigenvalue solver for the frequency dependency. The code has been checked against a spectral code using Chebyshev polynomials for the space discretization as well as a time evolution code using 4th order Runge-Kutta for the time and finite differences in space.

## Chapter 11

### Ionisations simulations using PIC-MCC

# Chapter 12

## Experimental diagnostics

To understand the behavior of the trapped distribution function I have created a list of quantities of interest:

- Currents and fluxes in all 4 boundaries
- maximum density
- Number of trapped particles
- Temperature
- 

Technology	Measured quantity	density	Limitations
Microwave interferometry	line integrated density	$> 10^{17}m^{-3}$	
Phosphor screen	radial distribution	$< 10^9m^{-3}$	$1W/cm^2$
Farraday cup	total charge		



# Chapter 13

## Future investigations

Now that we have a good confidence in the code numerical model and its implementation, we will be able to focus on the physic and to modify the simulation parameters. On this topic, the following suggestions have already been made:

- Study the Diocotron instability and find a criterion of stability allowing us to determine during or after the 2D simulation if a 3D simulation should be implemented and performed.
- Study the effect of an external electric field on relation [6.1](#)

# Appendix A

## Conservation of $H$ and $P_\theta$ in coaxial geometry

We consider a charged particle of a plasma cloud confined in a coaxial geometry with external magnetic and electric field. In this geometry an axial symmetry is considered  $\frac{\partial}{\partial \theta} = 0$ .

The electric potential  $\phi(r, z, t)$  is the sum of an external and self-consistent electric field:

$$\phi(r, z) = \phi^{ext}(r, z) + \phi^s(r, z, t), \quad (\text{A.1})$$

and the magnetic vector potential  $A_\theta(r, z)$  is only external.

The particle total energy  $H$  is in this case:

$$H = \frac{m}{2} (v_r^2 + v_\theta^2 + v_z^2) + q\phi. \quad (\text{A.2})$$

With  $m$  and  $q$  respectively the particle mass and charge, and  $(v_r, v_\theta, v_z)$  are the three components of the particle velocity. Due to the presence of the electric and magnetic field, we can write the equations of motions of the charged particle in cylindrical coordinate system.

$$m [\ddot{r} - r\dot{\theta}^2] = -q \frac{\partial \phi}{\partial r} + q \frac{r\dot{\theta}}{r} \frac{\partial r A_\theta}{\partial r} \quad (\text{A.3})$$

$$m [r\ddot{\theta} + 2\dot{r}\dot{\theta}] = -qz \frac{\partial A_\theta}{\partial z} - \frac{qr\dot{\theta}}{r} \frac{\partial r A_\theta}{\partial r} \quad (\text{A.4})$$

$$m [\ddot{z}] = -q \frac{\partial \phi}{\partial z} + qr\dot{\theta} \frac{\partial A_\theta}{\partial z} \quad (\text{A.5})$$

We

$$\frac{dH}{dt} = \frac{m}{2} \left( \frac{dv_r^2}{dt} + \frac{dv_\theta^2}{dt} + \frac{dv_z^2}{dt} \right) + q \frac{d\phi}{dt} \quad (\text{A.6})$$

$$= \frac{m}{2} \left( \frac{d\dot{r}^2}{dt} + \frac{d(r\dot{\theta})^2}{dt} + \frac{d\dot{z}^2}{dt} \right) + q \frac{d\phi}{dt} \quad (\text{A.7})$$

$$= \frac{m}{2} \left( 2\dot{r}\ddot{r} + 2(\dot{r}\dot{\theta} + r\ddot{\theta})r\dot{\theta} + 2\dot{z}\ddot{z} \right) + q \frac{d\phi}{dt} \quad (\text{A.8})$$

$$= m \left( \dot{r}(\ddot{r} - r\dot{\theta}) + (2\dot{r}\dot{\theta} + r\ddot{\theta})r\dot{\theta} + \dot{z}\ddot{z} \right) + q \frac{d\phi}{dt} \quad (\text{A.9})$$

Using equations [A.3](#) to [A.5](#) we get

$$\begin{aligned} \frac{dH}{dt} = & \dot{r} \left( -q \frac{\partial \phi}{\partial r} + q \frac{r\dot{\theta}}{r} \frac{\partial r A_\theta}{\partial r} \right) \\ & + r\dot{\theta} \left( -q\dot{z} \frac{\partial A_\theta}{\partial z} - \frac{q\dot{r}}{r} \frac{\partial r A_\theta}{\partial r} \right) \\ & + \dot{z} \left( -q \frac{\partial \phi}{\partial z} + qr\dot{\theta} \frac{\partial A_\theta}{\partial z} \right) + q \frac{d\phi}{dt} \end{aligned} \quad (\text{A.10})$$

## Appendix B

### Derivation of the self consistent electric field for an annular flat top density electron plasma trapped in an infinite length coaxial cylinder with biased electrodes

If we consider an annular electron plasma confined in a coaxial geometry as represented in figure B.1 with the following density:

$$n(r) = \begin{cases} 0 & a < r < r_b^- \\ \hat{n}_e & r_b^- \leq r \leq r_b^+ \\ 0 & r_b^+ < r < b \end{cases} \quad (\text{B.1})$$

and electric potential boundary conditions:

$$\phi(a) = \phi_a; \phi(b) = \phi_b; \phi(r_b^+) = \phi^+; \phi(r_b^-) = \phi^-. \quad (\text{B.2})$$

In this geometry we consider an infinite cylinder with  $\frac{\partial}{\partial z} = 0$  and  $\frac{\partial}{\partial \theta} = 0$ . We can therefore solve the poisson equation analytically in the three radial regions of the coaxial geometry:

$$\nabla^2 \phi(r) = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \phi}{\partial r} \right) = \begin{cases} 0 & a < r < r_b^- \\ -q\hat{n}_e/\epsilon_0 & r_b^- \leq r \leq r_b^+ \\ 0 & r_b^+ < r < b \end{cases} \quad (\text{B.3})$$

In the vacuum regions the electric potential can be written as:

$$\phi(r) = A^\pm \ln(r) + B^\pm \quad | \quad a < r < r_b^- \text{ or } r_b^+ < r < b, \quad (\text{B.4})$$

with  $A$  and  $B$  constants, and in the nonneutral plasma region:

$$\phi(r) = \frac{-q\hat{n}_e}{4\epsilon_0} r^2 + C \ln(r) + D \quad | \quad r_b^- \leq r \leq r_b^+. \quad (\text{B.5})$$

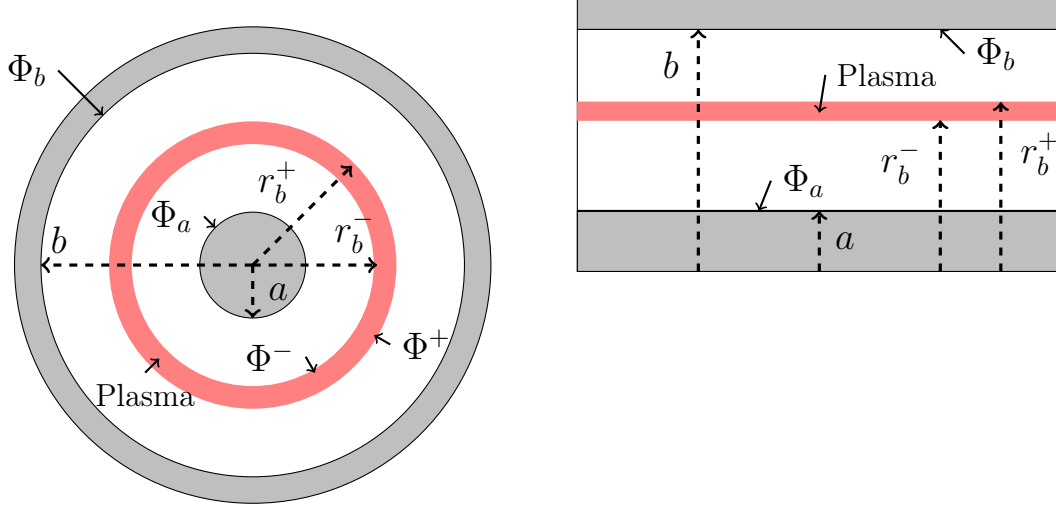


Figure B.1: Considered geometry seen in the  $(x, y)$  and  $(r, z)$  plane.

To calculate  $A^\pm$ ,  $B^\pm$ ,  $C$  and  $D$  we use the matching conditions at  $r_b^-$  and  $r_b^+$ .

$$\lim_{r \nearrow r_b^-} E_r = \lim_{r \searrow r_b^-} E_r \quad (\text{B.6})$$

$$\lim_{r \nearrow r_b^+} E_r = \lim_{r \searrow r_b^+} E_r. \quad (\text{B.7})$$

This means that:

$$A^- \ln(a) + B^- = \phi_a \quad (\text{B.8})$$

$$A^- \ln(r_b^-) + B^- = \phi^- \quad (\text{B.9})$$

$$A^+ \ln(r_b^+) + B^+ = \phi^+ \quad (\text{B.10})$$

$$A^+ \ln(b) + B^+ = \phi_b \quad (\text{B.11})$$

We can then solve for  $A^\pm$ ,  $B^\pm$ ,  $C$  and  $D$

$$A^+ = \phi_b - \phi_+ \quad (\text{B.12})$$

$$A^- = \phi_- - \phi_a \quad (\text{B.13})$$

$$B^+ = \phi_+ \ln(b) - \phi_b \frac{\ln(r_b^+)}{\ln(b/r_b^+)} \quad (\text{B.14})$$

$$B^- = \phi_a \ln(r_b^-) - \phi_- \frac{\ln(a)}{\ln(r_b^-/a)} \quad (\text{B.15})$$

$$C = \left[ \phi_+ - \phi_- + \frac{qn_e}{4\epsilon_0} ((r_b^+)^2 - (r_b^-)^2) \right] \frac{1}{\ln(r_b^+/r_b^-)} \quad (\text{B.16})$$

$$D = \frac{qn_e}{4\epsilon_0} (r_b^+)^2 + \phi_+ - C \ln(r_b^+) \quad (\text{B.17})$$

As well as calculate

$$\begin{aligned} \phi^+ = & \frac{1}{\ln(b/a)} \left[ \frac{qn_e}{2\epsilon_0} (r_b^-)^2 \ln\left(\frac{b}{r_b^+}\right) \ln\left(\frac{r_b^+}{r_b^-}\right) \right. \\ & \left. + \left( \phi_a + \frac{qn_e}{2\epsilon_0} ((r_b^+)^2 - (r_b^-)^2) \left( \ln\left(\frac{r_b^+}{a}\right) - \frac{1}{2} \right) \right) \ln\left(\frac{b}{r_b^+}\right) + \phi_b \ln\left(\frac{r_b^+}{a}\right) \right], \end{aligned} \quad (\text{B.18})$$

$$\begin{aligned} \phi^- = & \frac{1}{\ln\left(\frac{b}{a}\right)} \left[ -\frac{qn_e}{2\epsilon_0} (r_b^-)^2 \ln\left(\frac{r_b^-}{a}\right) \ln\left(\frac{r_b^+}{r_b^-}\right) \right. \\ & \left. + \left( \phi_b + \frac{qn_e}{2\epsilon_0} ((r_b^+)^2 - (r_b^-)^2) \left( \ln\left(\frac{b}{r_b^+}\right) + \frac{1}{2} \right) \right) \ln\left(\frac{r_b^-}{a}\right) + \phi_a \ln\left(\frac{b}{r_b^-}\right) \right], \end{aligned} \quad (\text{B.19})$$

and

$$\begin{aligned} \Delta\phi^{+-} = & \phi_+ - \phi_- = \frac{\ln\left(\frac{r_b^+}{r_b^-}\right)}{\ln\left(\frac{b}{a}\right)} \left[ \frac{qn_e}{2\epsilon_0} (r_b^-)^2 \ln\left(\frac{br_b^-}{ar_b^+}\right) + \phi_b - \phi_a \right. \\ & \left. + \frac{qn_e}{2\epsilon_0} ((r_b^+)^2 - (r_b^-)^2) \left( \ln\left(\frac{b}{r_b^+}\right) + \frac{1}{2} \left( 1 - \frac{\ln(b/a)}{\ln(r_b^+/r_b^-)} \right) \right) \right]. \end{aligned} \quad (\text{B.20})$$