

A MONTE CARLO CODE FOR SIMULATING THE NEUTRAL DENSITY PROFILE IN AN ECR ION SOURCE

Turn-in date: March 29, 2017 Supervisor: Taneli Kalvas

Abstract

In this research training project work, a Monte Carlo code was developed in order to estimate the spatial distribution of neutrals in the vacuum chamber of an ECR ion source. The ECR plasma was treated as a stationary collection of ion and electron populations, allowing to effectively reduce the computational complexity of the simulation and not having to simulate individual plasma particles. Despite the approximations, qualitative results corresponding to experimental observations were seen. For example, accumulation of neutrals in the ion loss zones was observed.

1 Introduction

In electron cyclotron resonance ion sources (ECRIS), it is crucial to keep the neutral density low enough to achieve efficient production of high ion charge states. The removal of neutrals is usually realized using turbo vacuum pumps. There are usually vacuum gauges installed in some locations to monitor the pressure.

The gauges, however, cannot capture local deviations in the neutral density. Therefore, the problem was approached by means of numerical simulation. The goal of this research training project was to produce a Monte Carlo code which could be used for that purpose.

It was assumed that the flow of neutrals is in the molecular flow regime, i.e. there is no direct interaction between the neutrals. The groundwork for the free molecular flow simulation was laid earlier in the author's BSc thesis [1].

A self-consistent ECRIS Monte Carlo simulation code was recently developed by Mironov [2]. That code is based on the Particle-in-Cell method and features a full dynamic model of the plasma ions and electrons. This kind of a model typically leads to great computational complexity, posing strong requirements on the choice of simulation hardware.

In this project the idea of a simpler, quasi-stationary plasma model was probed. Such a model has the advantage of lower complexity because one only has mutually non-interacting neutrals as test particles. The independence of the test particles also means that the computation is easily parallelizable and performance gains can be had from running on multiprocessor systems.

The bulk of the work in this project consisted of implementing a suitable plasma model and implementing neutral "ray tracing" in a real CAD geometry.

2 Theoretical background

In a typical ECR ion source, a plasma is formed by injecting neutral gas particles into a vacuum chamber and coupling RF energy to it at the frequency of several GHz.

The plasma in an ECR ion source is a typical example of a laboratory plasma, with an electron density of $\sim 10^{11} \text{ cm}^{-3}$. Due to the electron cyclotron resonance heating, there is a population of electrons with a high average energy of several keV. However, there is another population of electrons with lower energies. Thus, the electron energy distribution isn't strictly Maxwellian. [3]

Given that the ion confinement time in the magnetic bottle is long enough to achieve thermalization, the ion energy distribution can also be assumed as Maxwellian with a temperature of a few electronvolts [3].

2.1 Plasma processes

This section lists the collision processes where the plasma and neutrals are involved. The focus is purposely kept on reactions concerning the neutrals. Also, only the “single” versions of these processes are covered as these dominate over the multiple counterparts in most cases. The formulae for the respective cross sections or rate coefficients are also listed.

Using the cross section σ , the rate coefficient is then defined as $\langle \sigma(v_r)v_r \rangle$ where v_r is the speed of the target particle relative to the bombarding particles. Considering a population of bombarding particles whose velocities have a uniform direction distribution, the relative speed can be written as

$$v_r = \sqrt{v_x^2 + v_y^2 + (v_z - v_t)^2} \quad (1)$$

where v_t is the speed of the target particle and $v_{\{x,y,z\}}$ are the velocity components of the bombarding particle. Here the direction of the target particle motion was chosen to be along the z axis. This choice is arbitrary due to the symmetry of the velocity distribution of the projectile particles.

Now the rate coefficient can be calculated as an integral over the projectile particle velocity space:

$$\langle \sigma v_r \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v_r \sigma(v_r) f(v_x, v_y, v_z) dv_x dv_y dv_z, \quad (2)$$

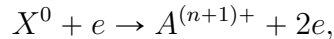
where $f(v_x, v_y, v_z)$ is the velocity distribution of the projectile particles. In the context of this simulation, f is the Maxwellian velocity distribution:

$$f(v_x, v_y, v_z) = \left(\frac{m}{2\pi kT} \right)^{3/2} \exp \left[-\frac{m(v_x^2 + v_y^2 + v_z^2)}{2kT} \right] \quad (3)$$

where m is the projectile particle mass, k is the Boltzmann constant and T is the temperature of the projectile particle population.

2.1.1 Electron ionization

Electron (impact) ionization is a process where an energetic electron positively ionizes a neutral by knocking off an electron bound to a shell:



where X is the neutral and e is an electron. Electron ionization of already positively charged ions is not considered here.

There have been many approximations of the electron ionization cross sections, most famously the Lotz equation [4]. A more convenient approximation was found in [5] and

Table 1: Fit parameters of equation 4 for argon neutrals [5].

| I (eV) | A | B_1 | B_2 | B_3 | B_4 | B_5 | condition |
|----------|-------|--------|---------|--------|-------|-------|-------------|
| 15.8 | 2.532 | -2.672 | 2.543 | -0.769 | 0.008 | 0.006 | |
| 15.8 | 4.337 | 3.092 | -21.253 | 14.626 | 0.018 | 0.031 | $E > 4.13I$ |

is given in the form of a fit function

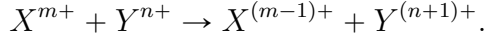
$$\sigma(E) = \frac{1}{IE} \left[A \ln \left(\frac{E}{I} \right) + \sum_{i=1}^N B_i \left(1 - \frac{I}{E} \right)^i \right] \quad (4)$$

where E is the electron energy, I is the ionization potential of the atom or ion and A , B_i are fit parameters depending on the atom or ion. The fit parameters are given in [5]. Notice that the cross section is given as a function of the electron energy instead of the relative speed — however, because the electron speed is undoubtedly much greater than the target particle speed, this is a perfectly valid approximation.

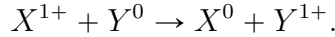
For example, the fit parameters for argon neutrals are listed in table 1.

2.1.2 Charge exchange

Charge exchange is a process where an ion X exchanges an electron with another ion or a neutral Y :



For example, X being a +1 ion and Y a neutral:



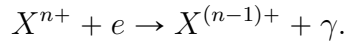
A famous approximation for the single charge exchange cross section is the one proposed by Salzborn et al. and used in [6] as

$$\sigma_{q,q-1} = 1.43 \times 10^{-12} q^{1.17} I^{-2.76} (\text{cm}^2), \quad (5)$$

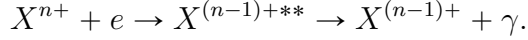
where q is the initial charge of the projectile and I is the ionization potential of the target particle in eV.

2.1.3 Recombination

In recombination, a free electron is bound onto a shell of an ion and the excess energy is released as a photon γ . If the release of energy happens “directly”, the process is called radiative recombination:



The released energy can also excite another bound electron which then de-excites and emits a photon:



This process is called dielectronic recombination. [7]

Precalculated rate coefficients for these two processes at various different electron temperatures were found at [8]. The rate coefficients of the two processes were summed to yield a total recombination rate coefficient.

2.2 Ion loss

In addition to recombination, a remarkable neutralization channel is the loss of ions from the magnetic bottle. When an ion is no longer confined by the magnetic bottle, it eventually hits a wall of the vacuum chamber. Inelastic collision reactions with the wall material eventually neutralize the ion and it will be adsorbed.

2.3 Knudsen's cosine law

The desorption of neutrals from walls of the vacuum chamber is assumed to be governed by Knudsen's cosine law [9]:

$$dn = \frac{n}{\pi} \cos \theta d\omega, \quad (6)$$

where n is the flux of particles desorbing into the solid angle element $d\omega$ and θ is the angle of $d\omega$ with respect to the surface normal.

2.4 Thermal accommodation

When a particle is adsorbed onto a wall, it may be only partially thermalized with the wall before desorption. The efficiency of this thermalization process is characterized by the *accommodation coefficient* α , which depends on the surface material, surface finish and the element of the incoming particles. It is defined according to the equation

$$\alpha = \frac{E_{\text{in}} - E_{\text{re}}}{E_{\text{in}} - E_{\text{w}}}, \quad (7)$$

where E_{in} is the incoming energy flux, E_{re} the reflected flux and E_{w} the emitted energy flux in the case of full thermalization. [10]

3 Numerical methods and tools

Several numerical methods and software packages were used to implement the calculation of the neutral density distribution. The actual implementation code can be found in a Git repository [11].

3.1 Monte Carlo test particle simulation

The basis of the simulation is the test particle Monte Carlo method. A limited set of test particles is chosen to represent the real particles and given a corresponding statistical weight in the simulation. Their phase space coordinates are randomly drawn from the corresponding probability distributions and propagated through the target geometry. On wall collisions, a new direction and velocity are calculated according to Knudsen's cosine law and the thermal accommodation law. The test particle simulation of neutrals without plasma is covered in detail in [1].

3.2 A simple quasi-stationary plasma model

The plasma model implemented in this simulation describes the plasma electrons and ions of different charge states as density distributions defined in a cubic grid. The velocities of the plasma particles are approximated as Maxwellian with predefined temperatures.

Taking the conservation of the plasma mass as a guiding principle, the following approximations were made:

- Stationarity: the plasma population parameters (density, temperature) are not affected by collision reactions.
- The spatial density distributions of the ion populations are proportional to the distribution of electrons. The code takes the global charge state distribution as an input.
- The plasma consists of only one element.
- Elastic collisions between the neutrals and the plasma particles are neglected.

An ionizing collision reaction may result in excess mass accumulating in the plasma. Conservation of mass is implemented here: the plasma must eventually emit a particle through some channel.

In this simulation, the two collision processes involving neutrals are charge exchange and electron ionization. In charge exchange, the plasma mass offset is unchanged because an ion is neutralized but the plasma also receives a new ion. Thus there is no need to emit

an additional neutral. In electron ionization, a neutral comes in and is ionized by an electron, thus the plasma gains an excess of one particle. The excess is compensated by statistically choosing a neutralization channel and the corresponding neutralization time.

3.2.1 Monte Carlo collisions

To handle collisions processes of neutrals with the plasma particles (ions and electrons), a Monte Carlo collision method has to be used. The electrons and ions don't exist in the simulation as individual test particles but as particle distributions.

Consider a test particle traveling through a particle population bombarding it at the frequency ν . Remembering that a single test particle represents multiple real particles, the propagation of the test particle can be seen as probability flux or an "intensity". The calculation of collision probabilities arises from the differential equation

$$\frac{dp}{dt} = -\nu p(t), \quad (8)$$

where p is the particle probability flux and t the time. Setting $p(0) = 1$, The solution to this differential equation is

$$p(t) = \exp(-\nu t).$$

The collision probability is now found as the complement:

$$p_{\text{coll}} = 1 - \exp(-\nu t), \quad (9)$$

where ν is the rate of the collision reaction in question. Considering a test particle propagating by a timestep Δt in a projectile particle cloud of density n_i and the rate coefficient of the reaction is $\langle v_r \sigma_i \rangle$, the probability of the collision happening during the timestep is

$$p_i = 1 - \exp(-n_i \langle v_{r,i} \sigma_i \rangle \Delta t), \quad (10)$$

where $v_{r,i}$ is the relative speed of the test particle to the projectile population. Drawing a pseudo-random number $x \in [0, 1[$, a collision now happens if

$$x \leq p_i = 1 - \exp(-n_i \langle v_{r,i} \sigma_i \rangle \Delta t),$$

or equivalently

$$x > \exp(-n_i \langle v_{r,i} \sigma_i \rangle \Delta t).$$

When there are multiple populations and collision reactions, the total collision frequency is calculated as:

$$\nu_{\text{tot}}(v) = \sum_i \nu_i = \sum_i n_i \langle v_{r,i} \sigma_i \rangle, \quad (11)$$

where i runs over different collision reactions. In practice, the values $\langle v_{r,i} \sigma_i \rangle$ are expensive to calculate and are thus precomputed in a grid of some fixed values of v and stored in an array. Linear interpolation between the gridpoints is used to acquire the collision frequency at arbitrary values of v .

The total collision frequency is used at each timestep to decide if a collision has happened. In the case of a collision, the collision reaction type i is randomly decided using the individual reaction rates as the probability distribution:

$$p[i] = \frac{\nu_i}{\nu_{\text{tot}}}. \quad (12)$$

Given a pseudo-random number $x \in [0, 1[$, the chosen reaction is the smallest value of i such that

$$x \leq \frac{\sum_{j=0}^i \nu_j}{\nu_{\text{tot}}}.$$

When the decision on the collision process is made, a projectile particle is sampled from the corresponding particle population. The collision dynamics are then calculated in the center-of-mass frame. The neutral product (if any) is then tracked further.

The reactions handled in the way described above in the simulation are charge exchange and electron ionization.

3.2.2 Neutralization channel sampling

If the plasma mass offset is increased due to an ionizing reaction, a particle will be emitted from the plasma and eventually neutralized. There is the possibility of specifying one or more neutralization channels, each of which follows an exponential decay law:

$$p_{\text{neutralized}}(t) = 1 - \exp(-t/\tau_i), \quad (13)$$

where $p_{\text{neutralized}}$ is the probability that the neutralization reaction has happened withing time t and τ_i is the time constant of the corresponding reaction. Given a pseudo-random number $x \in [0, 1[$, a random neutralization time can be sampled by conventional inverse transform sampling:

$$t = -\tau_i \ln x. \quad (14)$$

Because the neutralization neutralization processes are independent from each other, the following way of sampling is implemented:

1. For each neutralization channel, sample a random decay time.
2. Choose the reaction with the shortest decay time.

Practically, recombination and loss of an ion from magnetic confinement are the two neutralization processes. In recombination, the position of the reaction is sampled from the 1+ ion density distribution. The velocity of the resulting neutral is sampled according to the 1+ energy distribution (Maxwell).

The magnetic loss is a bit more complicated. Predefined “loss zones” calculated in an electron simulation are used. These are regions on the walls of the chamber where lost electrons have hit. The spatial behaviour of a plasma is usually dominated by the electrons and thus the ion loss zones should roughly correspond to the electron loss zones as seen in [12]. When an ion escapes through this channel, its position on the wall is chosen randomly according to the loss zones and it is assumed to be fully thermalized with the wall and neutralized due to recombination with wall electrons. Then a neutral is emitted from this position according to the cosine law.

3.3 Electron simulation

A simple, single electron simulation was used to calculate the electron distribution of the plasma. The electrons are simulated individually. Their trajectories are simulated in the target geometry which is approximated by a cylinder in the presence of a magnetic field. The magnetic field is given by the superposition of an axisymmetric expansion of the solenoid field and an ideal hexapole field. The single electron model is described in detail in [12].

The points where the electrons hit the chamber walls were recorded. Also, at each timestep, the locations of the electrons were recorded by incrementing the corresponding counters in a cube grid. From this grid of counters, the electron density was calculated by normalization. The choice of normalization is not exact. In the chosen approach, cells with counter value of 1/4 or more of the global maximum value were regarded as plasma cells. The normalization factor was chosen such that the average density in these cells evaluated to unity.

The coordinates of the electron endpoints corresponding to the “loss zones” mentioned earlier and the electron density distribution were stored as CSV data files.

3.4 Calculating the confinement time

Lastly, the confinement time of ions in the magnetic bottle had to be estimated. Geller [3] gives an equation for ions of charge state $q+$:

$$I_q \simeq \frac{1}{2} \frac{n_q q e \text{Vol}}{\tau_q}, \quad (15)$$

where I_q is the extracted ion current, n_q is the ion density, e is the elementary charge, τ_q the confinement time and Vol is the volume bounded by the magnetic field lines which lead to the extraction hole.

This equation can be solved for τ_q using measured extraction current. However, the volume Vol had to be estimated. A simple Monte Carlo approach was used here: N random positions were uniformly sampled on the extraction hole. The magnetic field lines

starting from these points were traced backwards until the chamber wall was intersected, flagging the cells of a cube grid through which the field line passed. The volume Vol was calculated as the sum of the volumes of these cells.

The magnetic field line backward tracing is implemented by numerically solving the differential equation

$$\frac{d\mathbf{s}(t)}{dt} = -\mathbf{B}(\mathbf{s}(t)), \quad (16)$$

where $\mathbf{s}(t)$ is the field line parametrized by the parameter t and \mathbf{B} is the total magnetic flux density.

3.5 Software design

One of the major challenges in the software design was efficiently calculating intersections of neutral paths with a CAD geometry with a non-trivial number of vertices and faces (given as STL files). The Computational Geometry Algorithms Library (CGAL) [13] was chosen as the provider of algorithms related to manipulation of triangulated meshes. This choice implied the usage of C++ as the programming language, which also made sense in terms of performance optimization.

The GNU Scientific Library (GSL) [14] was used for the computation of the rate coefficient integrals (2) and tracing the magnetic field lines as described in section 3.4.

The software library was designed with extensibility in mind. There are programming interfaces for moderately easy addition of new collision reactions and neutralization channels. Features of the modern C++14 standard such as smart pointers were used for safe handling of memory. This is a subject that could still be improved upon, especially regarding read and write access of the `std::vectors` storing the density distribution.

The neutrals were tracked individually one by one, sampling the particle position and incrementing the respective counter in a cube grid at fixed intervals. The implementation uses multiple threads allowing many neutrals to be traced simultaneously and increasing performance on multicore systems.

The output data was stored as CSV files for postprocessing and plotting.

4 Results

Time-dependent and stationary simulations were run to test the simulation code. A simplified solid CAD model of the upcoming JYFL ECRIS HIISI was produced using OpenSCAD [15] and used as the target geometry. Originally an STL exported from the full CAD model was tried. However, the intersection calculation was found to be too

expensive probably due to spatially imbalanced distribution and the great number of primitives.

4.1 Simulation parameters

Because as of writing this report there are no measurement of HIISI plasma parameters, some parameters from the JYFL 14 GHz ECRIS [16] were used. In particular, measured values of the extraction ion current of different charge states were used to obtain the charge state distribution. Additionally, the extraction current of $8+$ ions was used to calculate the confinement time τ_{8+} from equation (15). This confinement time was used as an approximation for all charge states. The calculation yielded a value of $\tau \approx 1.9$ ms.

The average electron density was assumed to be 10^{11} cm^{-3} . The electron energy distribution was taken to be Maxwellian, $T = 10 \text{ keV}$. This approximation neglects the cold electron distribution.

The element chosen for the gas feed was argon. Gas was fed to the chamber from the injection tube. The injection and extraction holes were configured as pumps with pumping efficiencies of 1 and 0.1, respectively.

4.2 Stationary density distribution

A stationary density distribution was calculated using the argon feed rate of $1 \text{ cm}^3 \text{ h}^{-1}$ in NTP conditions which is a typical value [17]. Ten million test particles were used. Slices of the neutral density distribution on different axes were plotted.

One of the main observations from the plots is the accumulation of neutrals near the loss zones in the grooves of the vacuum chamber and the injection and extraction ends. This corresponds to the loss of ions from the magnetic confinement and neutralization on the wall. The accumulation can be seen in figures 1 and 2.

The neutral density in the middle of the chamber (in the plasma region) seems to be at least an order of magnitude lower than the surrounding one, below 10^{10} cm^{-3} . This is due to the internal pumping effect caused mainly by the electron ionization process. The internal pumping makes it possible to meet the criterion for multiply charged ion production, $n_e/n_0 \gg 1$.

From the reaction statistics, it is additionally seen that only a negligible number of the neutralizations happen through recombination.

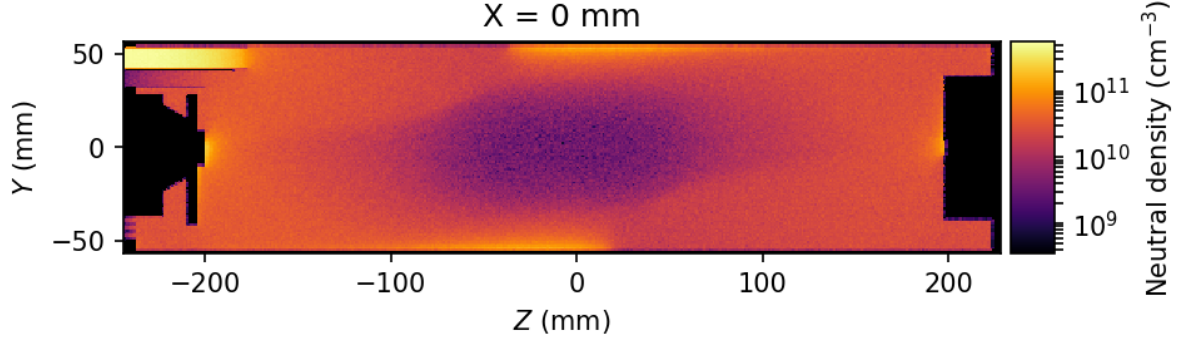


Figure 1: A slice of the stationary neutral density distribution in the middle of the X axis. The injection end is on the left, extraction end is on the right. Accumulation of neutrals can be seen on both ends and the cylinder walls in the middle.

4.3 Convergence test

To observe the convergence of the stationary neutral density distribution, the calculation was performed with several different numbers of test particles. The largest number was $N = 10^7$ and this result was taken as the “correct” distribution. The global error of the distributions obtained with lower number of statistics was then calculated using the ℓ^2 norm:

$$\delta = \sqrt{\sum_{i \in \text{grid cells}} (n_i - \hat{n}_i)^2}, \quad (17)$$

where \hat{n}_i is the neutral density in cell i in the reference distribution ($N = 10^7$) and n_i is the neutral density in the distribution whose error is being calculated.

The resulting data points and a line fit in $(\log N, \log \delta)$ coordinates are shown in figure 3. The slope of the fitted line is -0.44 ± 0.03 and tells that the power law governing the convergence rate is

$$\delta \propto N^{-0.44},$$

which coincides well with the known result that the convergence rate of a Monte Carlo integration method should be [1]

$$\delta \propto \frac{1}{\sqrt{N}}.$$

The power law is not exactly -0.5 but pretty close — this is due to the probabilistic nature of the Monte Carlo simulation. Therefore the convergence rate can be regarded as sensible.

4.4 Time dependent density distribution

Finally, a time dependent simulation with the same geometry and plasma parameters as the stationary one. A gas pulse lasting 1 ms consisting of 3×10^{12} physical particles was

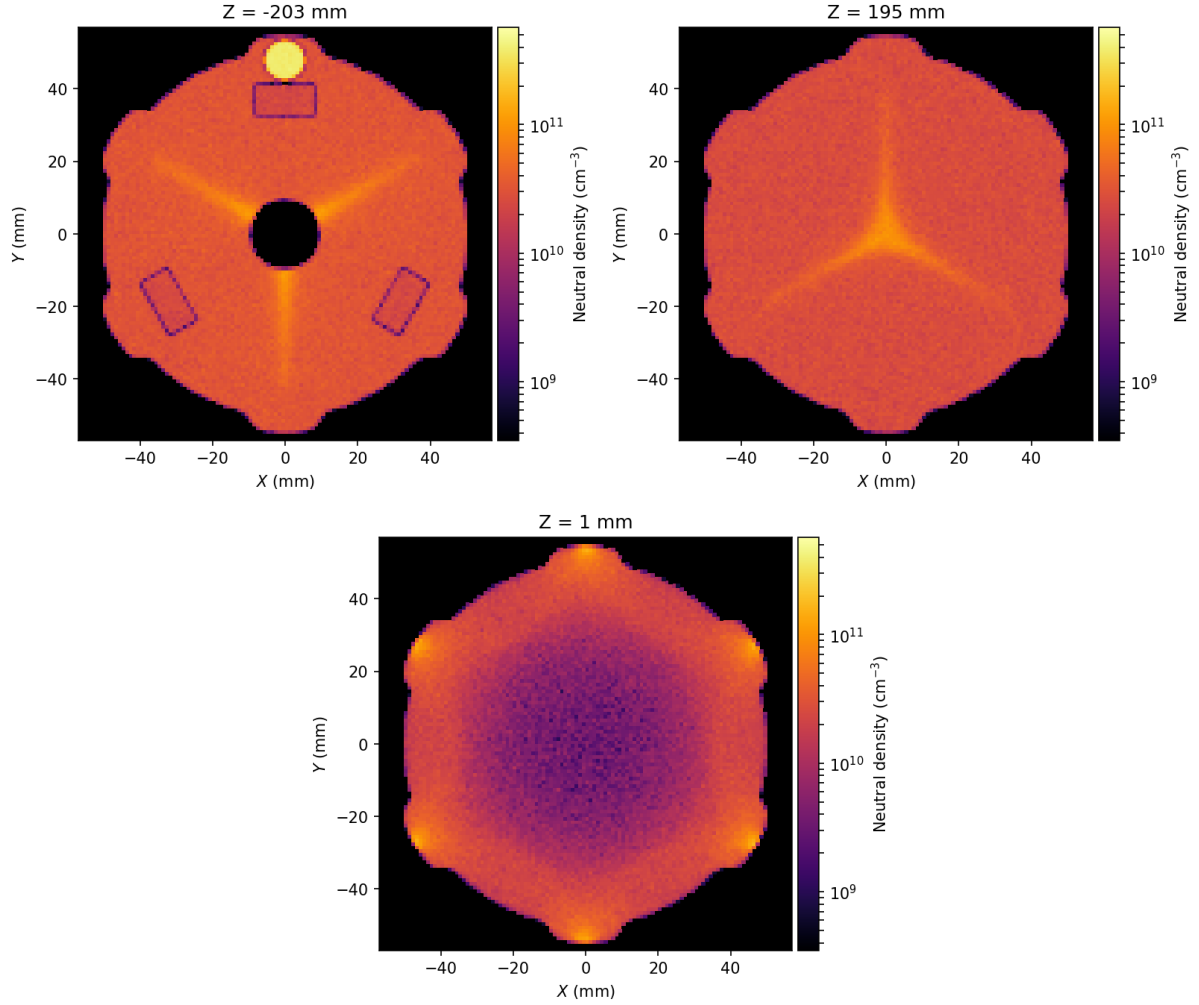


Figure 2: Slices of the stationary neutral density distribution along the Z axis at the injection end, extraction end and the middle of the chamber, showing the accumulation zones of the neutrals.

injected into the geometry via the injection tube, corresponding to some typical parameters used in a particular light diagnostic setup [18] developed at JYFL. 10^8 simulation particles were used for the simulation. The propagation of the gas pulse was tracked for 10 ms and sampled at 2 ms intervals. Some frames are plotted in figure 4.

The main observation from the time series is how the gas pulse is quickly diffused by the plasma reactions and wall collisions. The neutral density distribution then seems to settle and the neutrals are accumulated in the loss zones. These results coincide well with the quick rise and the slow decay of the light signal in [18]. If the simulation was continued further, all the particles would eventually be removed from the system through pumping channels.

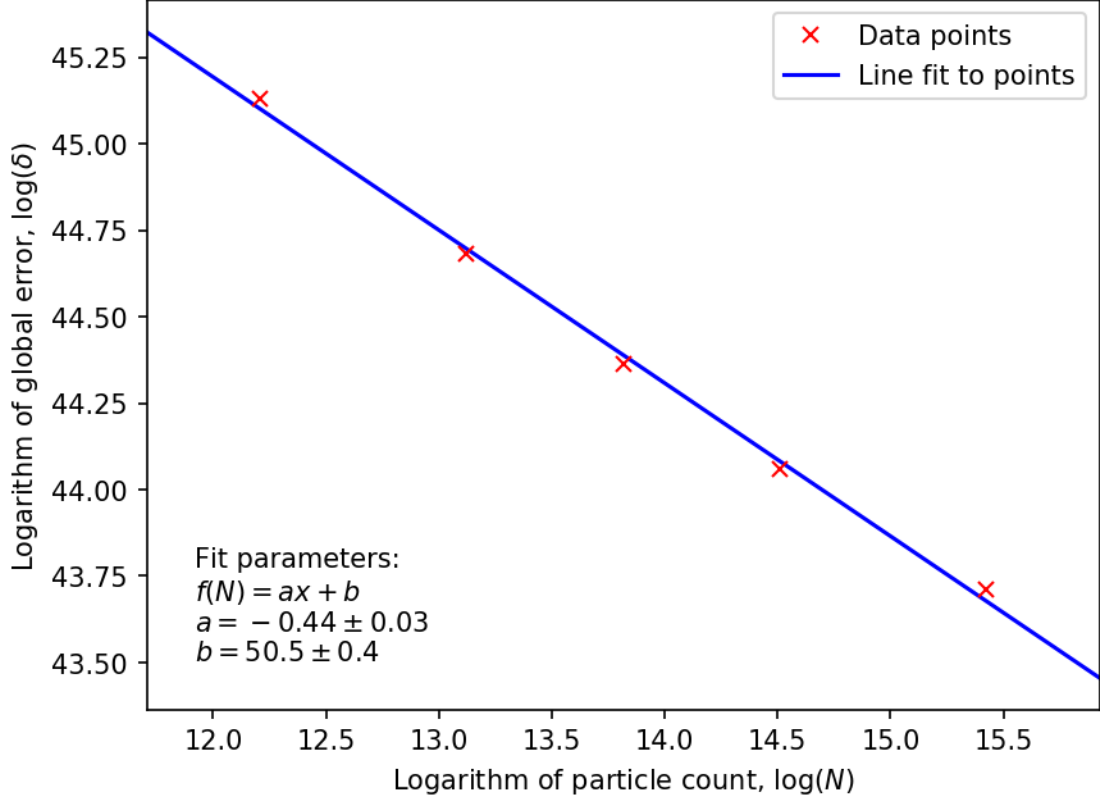


Figure 3: Global error plotted as the function of total test particle number and a line fit to the data points. In the log – log coordinates, the slope of the line equals to the convergence power law.

5 Conclusions

There were many approximations made when implementing the plasma model. In particular, neglecting the cold electron population is a significant approximation and may have caused the negligible presence of recombination reactions compared to other plasma processes.

It has been also observed in real ECR ion sources that the gas injection remarkably affects the plasma parameters. Thus the stationary plasma model used in this simulation may not hold valid especially in time-dependent simulations. In the stationary simulations the use of such a model is more justifiable, at least if the plasma parameters are measured in similar conditions (continuous gas feed).

Many simulation parameters such as the extraction efficiency coefficient and accommodation coefficient were left to be guessed. Therefore one must not interpret the simulation results too strictly — they are best taken as qualitative, order-of-magnitude descriptions of the distribution of neutrals in the vacuum chamber.

Comparing to a full-blown PIC model like Mironov’s [2], one disadvantage of this code

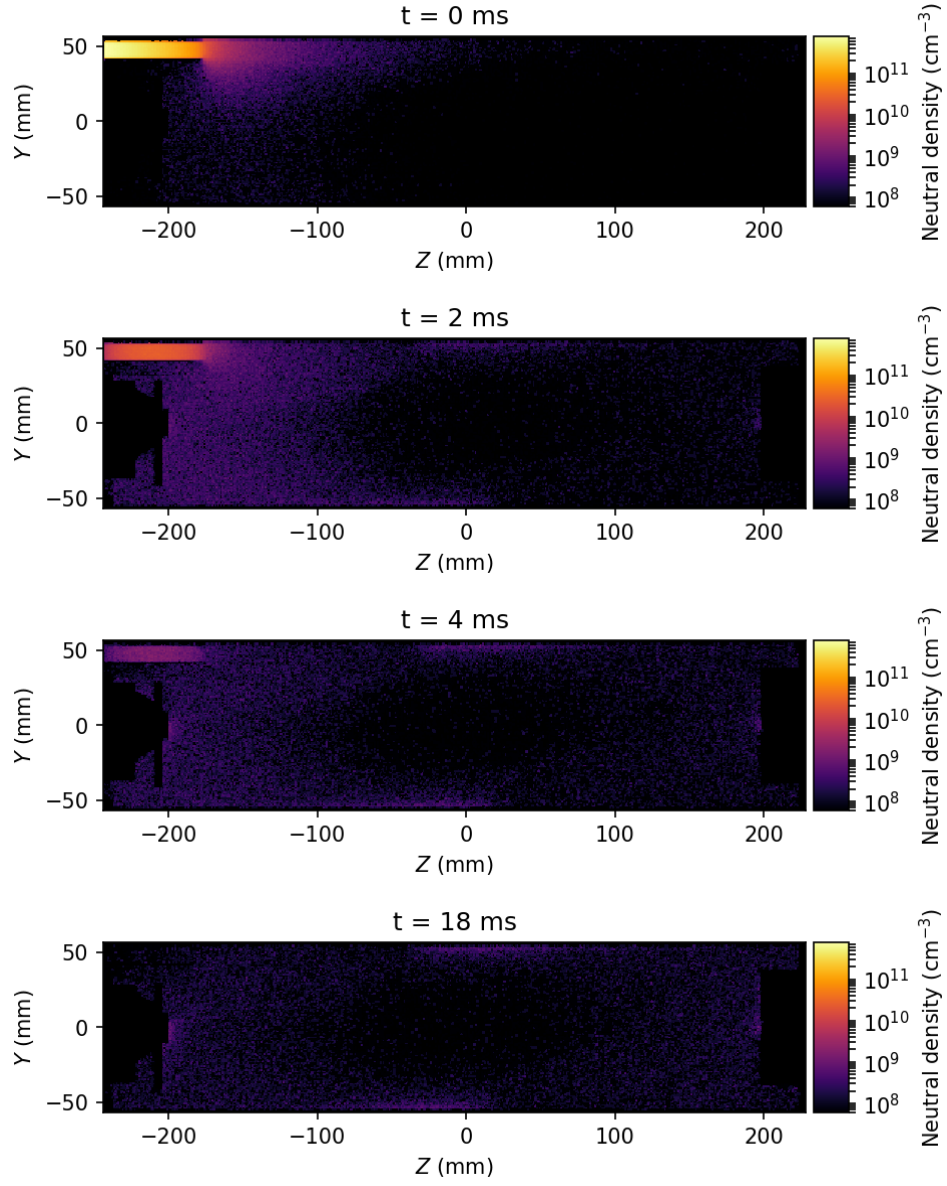


Figure 4: Results of the time dependent simulation.

besides the approximations mentioned above is the need for many input parameters like the charge state distribution. This is due to the fact that this code is not self-consistent. On the other hand, the model proposed in this research project is computationally less complex than a PIC model and run times are most certainly shorter.

Despite the approximations, the code managed to capture some interesting phenomena. First, the bombardment of ions on the loss zones seen in [12] is related to the localization of neutrals in the same zones in this simulation. Second, the quick rise and slow decay of the light signal waveform obtained in [18] corresponds to the time dependent results from the simulation.

Another strength of this simulation method is the scalability to multiprocessor systems. At least with the used HIISI geometry, the simulations could be run with sufficient statistics with reasonable durations, lasting 1-2 days per simulation run on a multicore laptop processor.

Due to the fact that the individual test particle histories are independent of each other in this model, the model can be considered linear: the net effect of several neutral sources can be found as the linear combination of the individual neutral density profiles. This means that, for example, the background neutral distribution caused by outgassing and leakage (see [1] for some estimates) could be precomputed and combined with a profile caused by gas injection to acquire the total distribution. This makes it possible to obtain an overall, qualitative picture of the relationships between the neutral densities in various locations in the geometry. This information could be used to deduce and estimate the internal neutral density based on readings of the vacuum gauges located at injection and extraction.

Developing a multiprocessor physics simulation code that handles large amounts of memory in C++ is not the easiest task one could tackle in a research training project. Not being in the know of various memory safety features introduced in the modern C++11 and C++14 standards beforehand, the author had to about learn these the hard way. After many fruitful debugging sessions and refactoring to embrace safer coding practices, the code works well and should be extensible with new processes quite easily.

References

- [1] S. Kapanen. Modeling the neutral density distribution of an ECR ion source. Bachelor's thesis, University of Jyväskylä, 2015. Only in Finnish.
- [2] V. Mironov et al. Numerical model of electron cyclotron resonance ion source. 2015.
- [3] R. Geller. *Electron cyclotron resonance ion sources and ECR plasmas*. Institute of Physics Publishing, 1996.
- [4] W. Lotz. An empirical formula for the electron-impact ionization cross-section. *Zeitschrift für Physik*, 206:205–211, 1967.
- [5] M. A. Lennon, K. L. Bell, H. B. Gilbody, J. G. Hughes, A. E. Kingston, M. J. Murray, and F. J. Smith. Recommended data on the electron impact ionization of atoms and ions: Fluorine to nickel. *Journal of Physical and Chemical Reference Data*, 17:1285–1363, 1988.
- [6] S. Bliman, S. Dousson, B. Jacquot, and D. Van Houtte. Charge exchange cross sections of argon ions colliding with rare gas targets at keV energies. *Journal de Physique*, 42 (10):1387 – 1391, 1981.
- [7] Radiative and dielectronic recombination. <https://www.nist.gov/pml/quantum-measurement/atomic-spectroscopy/radiative-and-dielectronic-recombination>. Referenced March 14, 2017.
- [8] FLYCHK rate coefficients. <https://www-amdis.iaea.org/FLYCHK/RATES/index.php>. Referenced March 14, 2017.
- [9] M. Knudsen. Das Cosinusgesetz in der kinetischen Gastheorie. *Annalen der Physik*, 48:1113–1121, 1915.
- [10] D. J. Rader, W. M. Trott, J. R. Torczynski, J. N. Castañeda, and T. W. Grasser. Measurements of thermal accommodation coefficients. Technical report, Sandia National Laboratories, 2005.
- [11] S. Kapanen. Simulation code Git repository. <https://yousource.it.jyu.fi/erkka/samakapa-erkka>. Referenced March 20, 2017.
- [12] T. Kalvas et al. Thermal design of refrigerated hexapole 18 GHz ECRIS HIISI. In *Proceedings of ECRIS-2014, Nizhny Novgorod, Russia*, pages 114–119, 2014.
- [13] The Computational Geometry Algorithms Library. <http://www.cgal.org/>. Referenced March 15, 2017.
- [14] GSL - GNU Scientific Library. <https://www.gnu.org/software/gsl/>. Referenced March 15, 2017.
- [15] OpenSCAD - The Programmers Solid 3D CAD Modeller. <http://www.openscad.org/>. Referenced March 17, 2017.

- [16] The University of Jyväskylä. JYFL 14 GHz ECRIS. https://www.jyu.fi/fysiikka/en/research/accelerator/ionsources/ecr2_html/.
- [17] O. Tarvainen. *Studies of electron cyclotron resonance plasma physics*. PhD thesis, Jyväskylän yliopisto, 2005.
- [18] R. Kronholm. Nopean kaasunsyöttölaitteiston, paineenmittauksen ja valodiagnostiikan kehittäminen ECR-ionilähteeseen. Master's thesis, Jyväskylän yliopisto, 2014.