UNIVERSITÁ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI FISICA E ASTRONOMIA "GALILEO GALILEI"

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CENTRO DI ATENEO DI STUDI E ATTIVITÁ SPAZIALI "GIUSEPPE COLOMBO"
CISAS

Corso di Laurea Magistrale in
Fisica

Development and improvement of a novel particle in cell code for plasma application in space and fusion

Giacomo Gallina

Scientific Advisors: Ch.mo Prof. D. Pavarin

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

UNIVERSITÁ DEGLI STUDI DI PADOVA

Dott. M. Manente

CISAS

UNIVERSITÁ DEGLI STUDI DI PADOVA

T4I S.r.l. SUPERVISOR

Scientific Supervisor: Ch.mo Prof. F. Marzari

DIPARTIMENTO DI FISICA E ASTRONOMIA "GALILEO GALILEI"

UNIVERSITÁ DEGLI STUDI DI PADOVA

Development and improvement of a novel particle in cell code for plasma application in space and fusion

---

Giacomo Gallina

September 26, 2016

Laurea Magistrale in Fisica

Abstract

> *Sometimes it is the people no one can imagine anything of who do the things no one can imagine*
>
> **— Alan Turing**

In the recent years the PIC simulations of plasma sources have become increasingly important to investigate nonlinear wave-particle interactions in space plasmas. In PIC simulations, individual particles are tracked in a Lagrangian frame in continuous phase space, whereas moments of the distribution such as densities and currents are computed simultaneously on Eulerian (stationary) mesh points.
Recently a new 3D PIC code has been developed at the University of Padua, named F3MPIC ([14],[30]).
In the present study we have developed a new version of the 3D PIC F3MPIC code, studying and validating new algorithms to manage the interactions between charged particles. In particular the following document is structured as follows: firstly we have introduced and validated a completely new Monte Carlo model to treat in a consistent and efficient manner the interactions between charged particles and neutral ones; secondly we have developed and validated a new "charge conserving" method in electromagnetic particle-in-cell simulations; in particular a first integration with a new electromagnetic solver, called ADAMANT, has been proposed. Finally a new high versatile particles tracking algorithm has been grown up and tested. In particular a new algorithm to manage secondary electrons emission has been introduced and made compatible with the new tracking. If compared with other computational models or with previous F3MPIC developed tools, the new algorithms are more efficient and highly innovative.
These new tools have been used to characterize a high-power (> 1KW) helicon plasma source (HPT) that is now in development at CISAS, a research group of Padua University. The presented work has been performed in collaboration with T4I S.r.l., which is a spin-off of the University of Padua.

Acknowledgement

> *You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. You have to trust in something - your gut, destiny, life, karma, whatever. This approach has never let me down, and it has made all the difference in my life.*
>
> — **Steve Jobs**
> (CEO Apple Inc.)

Desidero innanzitutto ringraziare il Prof. Daniele Pavarin per la fiducia che mi ha sempre accordato, il Dott. Marco Manente per la presenza, disponibilitá e capacitá di dirimere i miei dubbi, nonché per le numerose ore dedicate alla stesura di questo lavoro, e il Prof. Francesco Marzari per i preziosi consigli e per la costante collaborazione. Ringrazio inoltre il mio amico Matteo Vencato per le lunghe "chiacchierate" e i suggerimenti che in questi anni ci siamo sempre scambiati. Un sentito grazie va infine ai miei genitori, a mio zio Stefano e a mia zia Nadia, e soprattutto a mia sorella Camilla che mi ha sostenuto ed incoraggiato in ogni circostanza. Grazie a tutti, avete contribuito a rendere questo lavoro migliore.

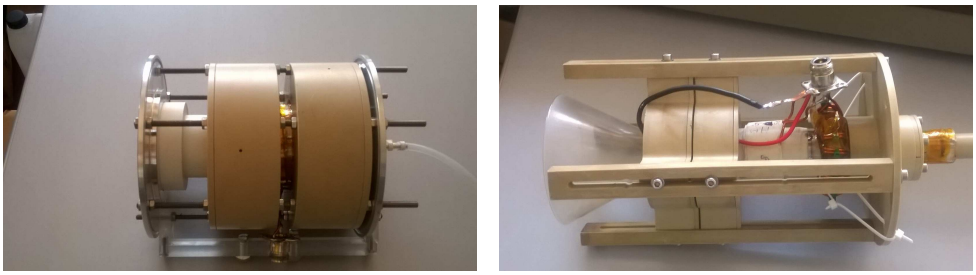
Contents

1 Introduction

> *Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.*
>
> **— Jules Verne**

Recently, plasma-based propulsion systems are beginning to challenge the monopole of chemical thruster in space applications. The high specific impulse, which allows for a huge reduction in the propellant mass, and high thrust efficiency make the plasma thruster an attractive solution for space propulsion. Whereas in a chemical rocket the specific impulse is an intrinsic characteristic related to the propellant calorific energy per unit of mass, in plasma propulsion systems, the specific impulse is extrinsic, depending on the electromagnetic energy deposited into the plasma.
Advances in plasma-based propulsion systems have led to the development of electromagnetic Radio-Frequency (RF) plasma generation and acceleration systems, called Helicon Plasma Thrusters (HPT). The HPT can be considered as an electrical propulsion system where the plasma is generated in a Helicon Plasma Source. A Helicon source consists of a dielectric tube surrounded by coils that generate a weak magneto-static field (up to 0.15 T) and a RF antenna working in the range of frequencies 1-50 MHz. The magnetic coils provide the quasi-axial magnetic field that allows the propagation of Helicon waves and the confinement of plasma inside the cylindrical source. Among different plasma sources, Helicon sources have been recognized as more efficient in depositing electromagnetic power and generating dense plasmas, in fact plasma density up to $10^{21}\ m^{-3}$ can be reached, using moderate magneto-static fields (below < 0.1 T).
Two projects that have developed the technology of the HPT are the European **HPH.COM** (Helicon Plasma Hydrazine Combined Micro), and the Italian **SAPERE** (Space Advanced Project for Excellence in Research and Enterprise).
The project HPH.COM aimed to develop a compact low-power plasma thruster using a high-efficiency plasma source based on helicon radio-frequency technology. The target applications are small satellites operating with an available propulsion power in the range of 50 W.
Otherwise in SAPERE project, a high-power (> 1 kW) plasma thruster is under development. The consortium for the design and realization of SAPERE is led by Thales Alenia Space Italia and involves also CISAS, a group of Padua University, as

one of the major partners. SAPERE is organized in two sub-projects: **STRONG** and **SAFE**.
SAPERE/STRONG aims at the realization of a reusable space tug coupled to the rocket VEGA for the transfer of payloads of different sizes from an intermediate orbit to the target orbit.



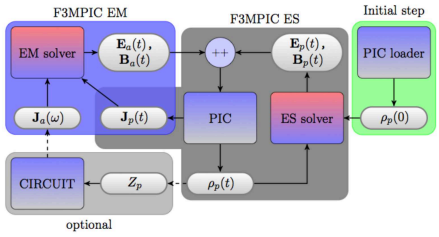
*(a)Prototype of the new thruster developed during the **STRONG** project*

(b)Other Radio-Frequncy thruster prototype

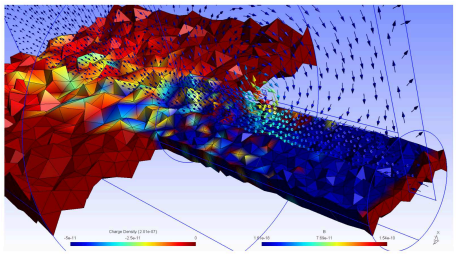
Fig. 1.1: *Left: Prototype of the new thruster developed during the STRONG project. Right: Other thruster prototype now in development at Cisas.*

The electric propulsion system will be a Helicon Plasma Thruster (HPT) with an input power higher than 1kW; the design of this thruster will be based on the scaling up of the prototype developed during the project HPH.COM.
Generally in a plasma-based thruster we can distinguish two main stages, where different physical processes take place: the production stage in the plasma source, and the acceleration stage at the exhaust section of the thruster.
In the production stage, plasma propulsion uses electric power to ionize the propellant and then imparts kinetic energy to the resulting plasma via energetic electron injection, biased electrodes or electromagnetic irradiation.
In the acceleration stage the plasma is exhausted by means of either electro-thermal, or electrostatic or electromagnetic processes. The more efficient the plasma generation and power deposition are, the better will be the HPT performance in terms of specific impulse and thrust efficiency. It follows that the HPT propulsive figures of merit are strictly related to the power deposited by the RF antenna into the Helicon source. Generally the physical processes that occur in a Helicon Plasma Sources are: plasma generation, wave-plasma coupling, and plasma transport. Within a helicon source, the plasma is also magnetized in order to enhance the lateral confinement and to permit the propagation of plasma waves (helicon and cyclotron waves) excited by the RF antenna. Differently than industrial helicon sources, a high kinetic energy must be delivered to ions. Furthermore, the plasma-wave coupling has to be optimized in order to maximize the ionization fraction. This makes necessary a deep understanding of the physical mechanisms involved, of both the electromagnetic coupling and the transport processes.
To investigate the required STRONG operational requirements, new numerical tools need to be developed to correctly design the above mentioned stages.

In the last years the PIC simulations of plasma sources have become increasingly important to investigate nonlinear wave-particle interactions in plasmas.
In PIC simulations, individual particles are tracked in a Lagrangian frame in continuous phase space, whereas moments of the distribution such as densities and currents are computed simultaneously on Eulerian (stationary) mesh points (see figure 1.2).
Recently a new 3D PIC code has been developed at the University of Padua, named F3MPIC ([14],[30]).
In its original implementation F3MPIC was developed for the detailed design and optimization of helicon and general-purpose plasma thruster and has been validated both numerically and experimentally under the HPH.COM project. F3MPIC has also been successfully applied to other plasma systems such as ion sources for the selective production of exotic species [27].



(a)F3MPIC general structure



(b)F3MPIC tetrahedral mesh with superimposed fields

Fig. 1.2: *Left: General scheme of F3MPIC structure. Right: General example of tetrahedral mesh used, with superimposed fields. This output example is made with an open source software called GMSH ([17])*

The code has been tested on High Performance Computing (HPC) facilities with a GPU version now under testing.
In F3MPIC, the classical PIC algorithm is coupled with a 3D finite element electrostatic solver in time called GETDP (see [12]). The code is built on an unstructured mesh of tetrahedra, allowing for arbitrary geometries, and the PIC core is comprised of a Boris-Leapfrog scheme that can manages multiple species, both charged and neutral. A magnetic field with an arbitrary topology can be imposed to study magnetized particle dynamics. Particles are tracked inside the tetrahedra using a fast priority-sorting algorithm and charge density is assembled on the nodes of the mesh at each time step (see figure 1.3).
The electrostatic fields are then computed by solving Poisson's equation or the full set of Maxwell's equations including both plasma source terms (charge density and plasma currents) and external source terms (e.g. a polarized electrode). The electrostatic field can be solved in either 2D axisymmetry or full 3D via a finite element method. Non-plasma regions (e.g. vacuum, conducting elements) may also be incorporated. Spatial distributions of plasma properties, electric and magnetic fields can be recorded at user-defined time steps. Arbitrary control surfaces can

record the positions and velocities of passing particles, allowing computation of distribution functions, mass flow rates, thrusts and beam divergence angles.

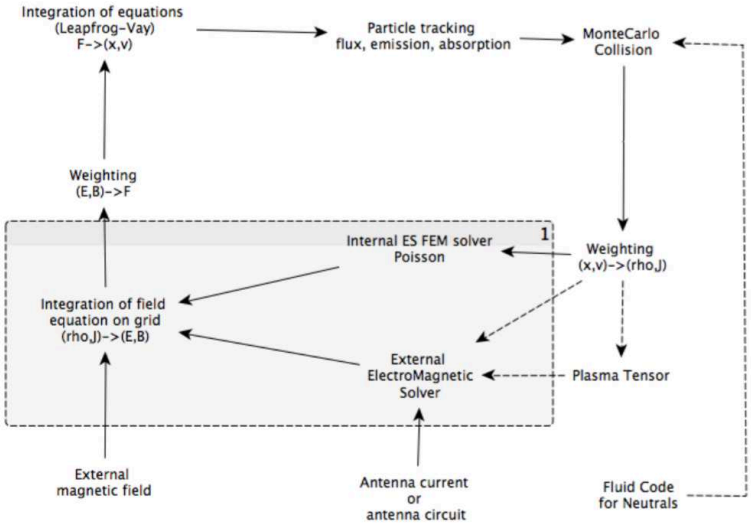


Fig. 1.3: *General scheme of F3MPIC structure. In this figure you can appreciate the main cycle used in F3MPIC with the two solvers. An electrostatic one and an electromagnetic one. The electromagnetic solver is now in development. See chapter 6. The figure has be taken from [15]*

F3MPIC has been validated on all its single parts involving its fundamental physics. In particular the electrostatic solver has been widely tested, both as a standalone Poisson solver, and coupled with the particle section of the code.
Some validation tests include: plasma sheath formation, diffusion time of charged species, the plasma drift subject to a constant electric field, etc. Also the particle mover has been widely tested, in order to estimate the errors on particle orbits, the stability of the calculated particle trajectories subject to simple and known forcing fields, the initial distribution of particle positions and velocities, and the reliability of the particle tracker during the transit between tetrahedra. The precautions that have been taken during the first phase of implementation have greatly reduced the number of loops searching needed to detect particles in an unstructured mesh, however algorithms working with advancing front are also easily prone to numerical errors due, sometimes, to the complexity of the geometry and these errors are difficult to predict and to categorize.
In the present study we have developed a new version of this 3D PIC code studying and validating new algorithms to manage the interaction between charged particles in such a way that a strong optimization of the existing STRONG hardware could be possible. The following thesis is structured as follows.
In the second chapter we have developed a completely new MonteCarlo code (MCC) to simulate interactions between charged particles and neutral ones. In particular,

starting from the work of V. Vahedi and M. Surendra [39], we have modified their scheme in order to take into account the effects of neutral pressure in a way that was not computational bundersome. A new recombination model has also been developed and integrated with the new MCC code. We have shown that our formulation is valid for arbitrary neutral pressure and at low pressure value is compatible with V. Vahedi and M. Surendra's model. Our model is also compatible with arbitrary grid and arbitrary tracking method.

In the third chapter we have implemented a new charge conservation method fully compatible with standard PIC structure following Umeda's paper (see [38]). After the development of a background framework to explain in detail why this new formulation preserves the continuity equation, we have developed a new structured mesh needed to deposit the current density vectors. The basic idea is to combine two meshes: a cubic one in which the current deposition is made, and an unstructured one, in which the field integration phase and the subsequent advancement of the particles are managed.

In the fourth chapter we have implemented and tested a new particles tracking algorithm proposed recently by Haselbacher A. and others [18]. If compared with previous F3MPIC tracking method, the new one is more efficient. In this chapter we have also shown a completely new boundary management algorithm to manage boundary crossing, internal deposition and secondary electrons emission.

In the fifth chapter we have shown the results of some simulations obtained using the new developed tools. The objective of these simulations is to test the new HPT thruster that is now in development at CISAS. In particular we have performed a deep analysis of the involved electromagnetic fields. A characterization of the source nowadays is not feasible, in fact the real experimental plasma density is very high $\sim 10^{19} m^{-3}$ and, at this density, it is difficult to have sustainable computing times using a PIC code that runs on a single processor. In the months to follow, we will complete the parallelization of F3MPIC and the new code will be used to conclude the analysis started in this chapter.

Finally in the six and last chapter we have introduced some new numerical tools to integrate the old electrostatic F3MPIC solver called GETDP, with a new electromagnetic one called ADAMANT. The effective integration requires a more detailed study that will be addressed in the following years. If compared with previous F3MPIC algorithms or with others well established numerical tools, the new algorithms are generally faster and more efficient.

This new version of F3MPIC will be intensively used; in particular a first strong experimental campaign will be done with the aim to validate, also experimentally, the new code. In such a way the new developed tools will be at the basis of a plasma based research whose objective is to continue the study and the optimization of a customized high-power plasma source.

2

A new Monte Carlo collision model to take into account neutral pressure variation and ion recombination, compatible with standard PIC formulation

> *A computer would deserve to be called intelligent if it could deceive a human into believing that it was human.*
>
> **— Alan Turing**

Abstract

In order to use particle-in-cell (PIC) simulation codes for modeling collisional plasmas, it is necessary to add interactions between charged and neutral particles. For this reason a detailed description of collision models is of the highest importance in Monte Carlo simulations.
Starting from the work of V. Vahedi, M. Surendra [39] it is necessary, for our purpose, to modify their scheme in order to take into account the effects of neutral pressure in the study of collisional events between charged particle and neutral ones. To do this we have modified classical PIC scheme considering particle mean free path. In this chapter we apply this new scheme using three major interaction models to compute mean free path: hard sphere interaction, screened Coulomb's interaction (see [16]) and modified hard sphere interaction. The choice depends on the type of interactions considered possible. We show that our formulation is valid for arbitrary neutral pressure and, at low pressure value, is compatible with V. Vahedi, M. Surendra's model. A new model for ion recombination was also developed and made compatible with the new algorithm. Our model is also compatible with arbitrary grid and arbitrary tracking method.

2.1 Introduction

Electromagnetic particle in cell (PIC) code are widely used for studies of interaction in plasma. A thorough description of the PIC technique can be found in Birdsall and Langdon's work [9]. In the last twenty years different collisional plasma models have been proposed to obtain realistic simulations to model plasma interactions with neutral particles. In 1994, V. Vahedi and M. Surendra [39] introduced a simple algorithm to simulate plasma interactions. The principale advantage of their scheme is that is fully compatible with PIC structure and thus, generally, easily implementable. Assuming to work in a uniform density distribution of neutral particles, they introduced a new collisional process, called *null collision* and they used it to find, statistically, the number of particles that undergo collision. The effective particles that collide are chosen randomly, and for each of them the type of the collision is checked.

The main disadvantage of their scheme, omitting the uniform pressure condition that in most practical situations is well verified, is closely connected to the need in the various MCC code to have one collision per particle per time-step. This request, in a situation in which the density of neutral particles is relatively high[1], brings to the need to choose a very small simulation time-step, which can lead to have unsustainable calculation times.

For this reason it was necessary, for our purpose, to modify their scheme in order to take into account the effects of neutral pressure variation in a way that was not computationally burdensome. A new model for ion recombination was also developed and made compatible with the new algorithm.

The model that we propose has the advantage to maintain the time step fixed and to work with particle mean free path to find the time between two following collisions inside each time-step. In this way it is not necessary to modify global time step to take into account pressure variations. The type of particle collision is always chosen randomly in accordance with Vahedi and Surendra's work.

In the presented scheme, the only approximation that we have done is connected with the fields updating; strictly speaking we have requested that the fields variations, due to particle motion during the same global time-step, are negligible if compared with the same variations calculated at any global time-step. This approximation is valid only in a system to convergence i.e. after the initial transient. However, if the code has a dump file, (i.e. the possibility to change run-time simulation time-step) the algorithm is also applicable in the initial transient phase, choosing a compatible time-step.

We conclude this introduction stressing that the choice of the particle mean free path is closely related to the choice of the possible particles interactions.

In this chapter we describe our scheme using three major interaction models to

[1] Vacuum medium regime

compute mean free path: hard sphere interaction, screened Coulomb's interaction (see [16]) and modified hard sphere interaction.
This chapter is organized as follows: firstly we recall the Vahedi and Surendra's model with an emphasis on its limitation; secondly we propose a variation of it using particle mean free path to find a different formulation valid in a general framework; especially we will show that our model, in a condition of low gas pressure, is exactly compatible with the first version of V. Vahedi, M. Surendra's work. Thirdly we conclude with some simulation results.
All the theory and computations are performed in an uniform neutral pressure condition but, as it will be shown, our model can be easily extended in a straightforward manner in a non-uniform neutral pressure situation.

2.2 Vahedi and Surendra's model

In this section we briefly show the main features of Vahedy and Surendra's work in a new way, following a statistical approach.
Assume that the particle species s has N types of collisions with a target species. If we consider the particle i with energy E_i, the total collision cross section $\sigma_T(E_i)$ is the sum of the N cross sections. We denote with $\sigma_j(E_i)$ the cross section of the j-th type of collision between the s species and the target species. In this way $\sigma_T(E_i)$ is

$$\sigma_T(E_i) = \sigma_1(E_i) + \sigma_2(E_i) + ... + \sigma_N(E_i) \tag{2.1}$$

Assume now to consider a single particle i, in the following called: the bullet, of velocity $\mathbf{v_i}$, that at most can move to a length Δs for a macro-time step Δt, i.e. $\Delta s = |\mathbf{v_i}|\Delta t$. Divide now the interval Δs into many intervals, each of length dx and assume that the probability of collision of the bullet in each of these intervals is proportional to dx itself. Assume now that the probability that in dx occurs more than one events, is negligible in comparison to that which it occurs exactly one.
In such a way we can calculate the probability that a collision occurs in the interval dx, i.e. dp, in the following way

$$dp = \sigma_T(E_i) n_t dx \tag{2.2}$$

where n_t is the density of the background species.
From this equation it is clear that the probability of collision, in the space unit dx, is given by $t \equiv \sigma_T(E_i) n_t$. If we assume that t is constant in each of the intervals dx[2] of Δs, then we have N_{tot} intervals ($\Delta s = N_{tot} dx$) in which we are interested only in the number of collisions. It follows that the average number of collisions in a length

[2]We are neglecting the dependence of $\sigma_T(E_i)$ by the energy of the particle; in other words it is as to consider particles with a constant energy E_i and thus a constant velocity $v_i \equiv |\mathbf{v_i}|$ in the interval Δs. This aspect can be easily accepted choosing a small enough thickness Δs.

Δs is given by a binomial distribution with expectation value $N_{tot}\sigma_T n_t \Delta s$.
This aspect can be easily seen in a different way.
Let's consider N independent particles, and let's call dp the probability for a particle to collide. Assume now, as before, that this probability is independent on the particle taken into consideration, and assume that the working conditions of the note 2 are still valid.
This phenomenon is always described by a binomial distribution and the average number of collisions of N particles, in a length Δs, is always given by $N\sigma_T n_t \Delta s$.
From this we can see that if we consider a length dx, the number of particles that have not yet collided is reduced of

$$dN = -Ntdx \tag{2.3}$$

Separating the variables and integrating, the average number of particle that have moved for a length Δs, without colliding is given by

$$N(\Delta s) = N_0 e^{-t\Delta s} \tag{2.4}$$

where N_0 is the number of independent particles at the beginning. From this follows that, at Δs, the number of particles that have collided is given by

$$\tilde{\mathrm{N}} \equiv N_0 - N(\Delta s) = N_0 - N_0 e^{-t\Delta s} = N_0(1 - e^{-t\Delta s}) \tag{2.5}$$

In this way it is clear that, fixing a particle i of velocity $v_i \equiv |\mathbf{v_i}|$, the probability P_i for this particle to undergo collision is $1 - e^{-t\Delta s} = 1 - e^{-tv_i\Delta t}$.
To connect this result with Vahedy and Surendra's model, it is now necessary to compute the relative frequency of occurrence of any possible process for particles collision.
To do this we follow Vahedy and Surendra's article, in particular we introduce a new collision process with a collision frequency ν' given by

$$\nu' = \max_{\mathbf{x}}[n_t(\mathbf{x})]\max_{E}[\sigma_T(E)|\mathbf{v}(E)|] \tag{2.6}$$

which, when it is added to the total collision frequency $n_t(\mathbf{x})\sigma_t(E)|\mathbf{v}(E)|$, gives a constant value over all $\mathbf{x}$ and E i.e. over all positions and energies. This collisional process is called the null collision since no real interaction occurs. In this way the maximum fraction of the total number of particles in the simulation $\tilde{\mathrm{N}}/N_0$ which experience collision is given by

$$P_{null} = 1 - \exp(-\nu'\Delta t) \tag{2.7}$$

From our demonstration, it is now clear the origin of this probability: P_{null} follows directly from equation 2.5. However this new formulation is strictly valid only because

ν' gives a constant value over all $\mathbf{x}$ and E in a fixed time-step, as the t factor previously introduced. In other words, this P_{null} could be considered as the probability associated to a new collision process characterized by a constant collision frequency and therefore a constant probability.
Took note of this, the effective particles that collide are chosen randomly, while the specific interaction processes are chosen simulating a random number $R \in [0, 1]$, and choosing a specific collision process as follows

$$\begin{aligned} R &\leq \nu_1(E_i)/\nu' && \text{Collision type 1} \\ \nu_1(E_i)/\nu' < R &\leq (\nu_1(E_i) + \nu_2(E_i))/\nu' && \text{Collision type 2} \\ &.... \end{aligned} \tag{2.8}$$

where ν_j is the collision frequency of j-th type of collision for particle i, i.e.

$$\nu_j = n_t \sigma_j |\mathbf{v_i}| \tag{2.9}$$

After collision, particles are advanced using a suitable integrator.
The main advantage of this scheme is that it is fully compatible with PIC scheme and easily implementable; however it presents some disadvantages that, for our purposes, it was necessary to solve.

- Firstly, it is valid only in an uniform pressure condition indeed, only in this way, the random choice of colliding particles is completely satisfied

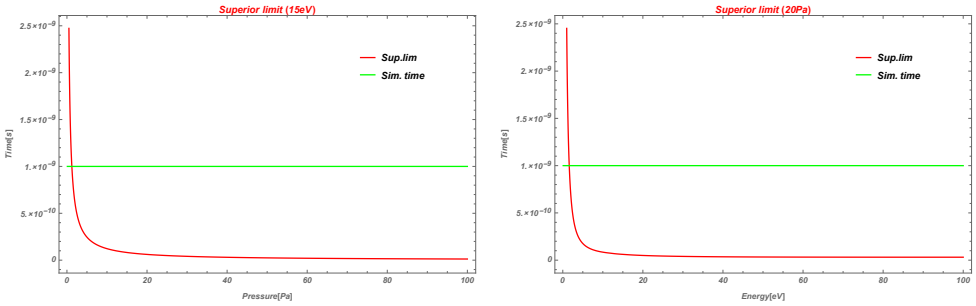
- Secondly, in a condition of relative high pressure, the choice of the time step Δt is computationally expensive and, practically, only really usable in a high vacuum regime.

To see this second aspect in great detail, it is necessary to analyze two things: the origin of P_{null} and the condition that Vahedy and Surendra proposed for the choice of the global time step Δt. Let's start with the origin of P_{null}.
In this section we have demonstrated as P_{null} follows directly from condition 2.5; this condition is only valid if we assume to have a group of independent particles that can undergo collision and if, and only if, we have at most one collision per particle per time-step. If this is not true, the whole construction loses meaning and the calculation of the number of particles that undergo collision, (i.e. $\tilde{N}$), is no longer correct. This aspect is closely linked to the second important analysis: the choice of the global time step Δt of the simulation.
In fact, to avoid to have multiple collisions in the same time-step for the same particle, Vahedy and Surendra proposed to choose the global time-step in such a way the probability P_i for each particle i to have a collision, is less than 0.01. In this way, being already very unlikely for a particle to have a single collision, it has averted for it to incur in multiple collision events.

This need can be obtained, as they suggested, given a uniform background gas density n_t, choosing Δt in a such a way that for particle i is valid the following condition

$$\Delta s_i \sigma_T(E_i) n_t \leq 0.1 \longrightarrow \Delta t \leq \frac{0.1}{v_i \sigma_T(E_i) n_t} \qquad \forall i \in \text{Particles} \tag{2.10}$$

using $\Delta s_i = v_i \Delta t$.
However this condition is very difficult to implement in a consistent way. In fact it should be necessary, after each particle motion and for each particle, to compute, using 2.10, an upper limit to the choice of the global time step Δt, and to compare all the computed Δt with each other, in order to choose only one global time step that fulfills 2.10 for all particles.



(a)Superior time-step distribution in function of the neutral pressure

(b)Superior time-step distribution in function of the electron energy

Fig. 2.1: *In figure a) and b), it is shown the trend of the superior time-step defined in 2.10 compatible with Vahedy and Surendra's model. In figure a) this limit is plotted in function of the neutral pressure (measured in Pa) keeping the electron energy fixed at $15eV$, while in figure b) this limit is plotted in function of the electron energy (calculated in eV) keeping the neutral pressure fixed at 20Pa. In Figure a) and b) we have also indicated, with a straight green line, the chosen time-step, that, in this configuration is of $1ns$. It is evident from figure a), considering a pressure of $20Pa$, that this time-step is not compatible with the superior time-step limit.*

In this way time-step is not necessary fixed for all the simulation, indeed it may have to be changed because it depends on the energy of the various particles, that is strictly connected with the configuration of the electric and magnetic fields.
In figure 2.1 it is shown the trend of this limit. It is clear that, already at neutral pressure of 20 Pa, it is necessary to choose a very small time step to satisfy condition 2.10 and thus the simulation would be very computationally expensive.

2.3 A new formulation to obtain a consistent pressure variation model

2.3.1 Connection of the standard PIC model with mean free path

Starting from this section we present our formulation.
To explain it in great detail we will proceed by steps. The crucial point is to untie the global collision time step Δt from the density of neutral particles. To do this we have developed a new model in which a crucial role is the use of the mean free path to estimate the time τ between two subsequent collisions.
The mean free path can be considered as the average distance between two collisions in a gas. In the following we will denote it with: λ. The explicit form of λ depends, among other things, on the density of the target species but also on the type of interactions that there may be between the particle (i.e. the bullet) and the neutral background (i.e. the target).
If we model the gas particles as hard spheres (non overlapping spheres), the expression for the mean free path (see [16]) is given by

$$\lambda = \frac{1}{\pi D^2 n_t} \tag{2.11}$$

where $D = R_{target} + R_{bullet}$ with R_{target} and R_{bullet} radius of the "sphere" representing respectively the target and the bullet.
This formulation is only valid if we assume elastic scattering between the target and the bullet and if we consider the target atoms fixed (not in motion).
Leaving out the first request that is closely related to the chosen model, the last is not well tested physically.
For this reason it is necessary to change equation 2.11 to take into account the motion of the atoms of the target. Keeping the interaction between the projectile and target atoms as a hard sphere interaction, i.e. purely geometrical, equation 2.11 can be modified as follows [16]:

$$\lambda_{\text{bullet}}[\text{ cm}] = \frac{s}{(s+\frac{1}{2s})\text{erf}(s) + \frac{1}{\sqrt{\pi}}\exp(-s^2)} \times \frac{3.297\text{cm} \times T[K]}{(R_{bullet}[\text{pm}] + R_{target}[\text{pm}])^2 P_{gas}[\text{mbar}]} \tag{2.12}$$

with s defined as

$$s = 107.7242\sqrt{\frac{E[eV]}{T[K]}}\sqrt{\frac{M_{target}}{M_{bullet}}} \tag{2.13}$$

To represent, as realistic as possible, the interaction between the charged species and the neutral gas in a plasma, it is however necessary to discard the hypothesis of hard sphere interactions. Indeed, like in [39], it is necessary to introduce different types of collisions.

Precisely for the "bullet" electrons we have considered the following types of collisions:

$$(1) \quad e + Ar \longrightarrow e + Ar \qquad \text{Elastic Scattering} \qquad (2.14)$$

$$(2) \quad e + Ar \longrightarrow e + Ar^* \qquad \text{Excitation} \qquad (2.15)$$

$$(3) \quad e + Ar \longrightarrow e + Ar^+ + e \qquad \text{Ionization} \qquad (2.16)$$

and for the "bullet" ions:

$$(4) \quad Ar^+ + Ar \longrightarrow Ar + Ar^+ \qquad \text{Charge Exchange} \qquad (2.17)$$

$$(5) \quad Ar^+ + Ar \longrightarrow Ar^+ + Ar \qquad \text{Elastic Scattering} \qquad (2.18)$$

Starting from these reactions, we have modified equation 2.11 to consider cross sections of these process.
Especially for a charged electron of energy E_i we have considered a mean free path of this type [26]

$$\lambda_{\text{elec}} := \frac{1}{n_t \sigma_{\text{Tot elec}}(E_i))} = \frac{1}{n_t(\sigma_1(E_i) + \sigma_2(E_i) + \sigma_3(E_i))} \qquad (2.19)$$

while for a charged ion

$$\lambda_{\text{ion}} := \frac{1}{n_t \sigma_{\text{Tot ion}}(E_i))} = \frac{1}{n_t(\sigma_4(E_i) + \sigma_5(E_i)} \qquad (2.20)$$

where σ_l with $l = 1...5$ are the cross sections of the previous reactions.
With the help of equation 2.19 and 2.20, we are able to determine the collision frequency of electron and ion collisions.
For example let's consider an electron or an ion of velocity $\mathbf{v_i}$, and consider λ_i as the corresponding mean free path.
The collision frequency is

$$\nu_i = \frac{|\mathbf{v_i}|}{\lambda_i} = \frac{\sqrt{v_{ix}^2 + v_{iy}^2 + v_{iz}^2}}{\lambda_i} \qquad (2.21)$$

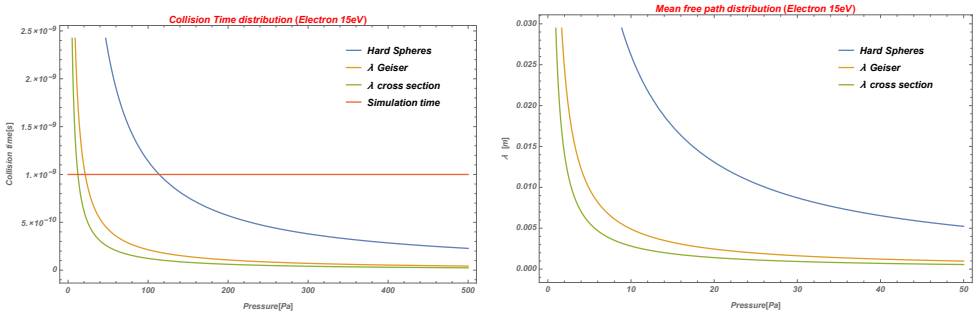
Now, using 2.21, we can compute the time interval between two following collisions as:

$$\tau_i = \frac{1}{\nu_i} \qquad (2.22)$$

Equation 2.22 is only valid if the target species is considered fixed. To consider neutral motion, it is possible to modify equation 2.22 in the following way (see [16])

$$\tau_i = -\frac{\ln(r)}{\nu_i} \qquad (2.23)$$

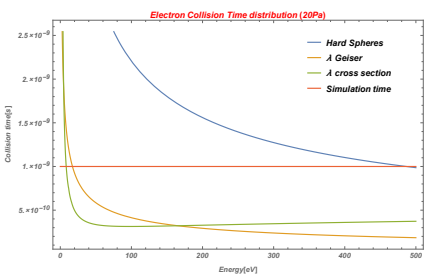
where r is a random number from a uniform distribution between zero and one.
In figures 2.2b) we have shown the trend of the the mean free path obtained using the three different formulations proposed in equations 2.12,2.19 and 2.11, while in figure 2.2a) we have reproduced its associated collision time easily obtainable from 2.22.
In these figures we have considered electrons as "bullets" and the Argon neutrals as the "target" species. In particular in figure 2.2b), the mean free path has been computed for electrons with an energy of $15eV$. This energy is the thermal agitation energy associated to an electron temperature of $116000K$.
We want already to point out as, in the model proposed by us, there isn't an upper limit or other prescription on the value of global time step Δt; however, as in standard PIC theory, the global time-step can not be taken arbitrarily large. In fact it must be compatible with mesh size and other computational theorems (for example Debye's theorem). See [9],[40],[13],[19].



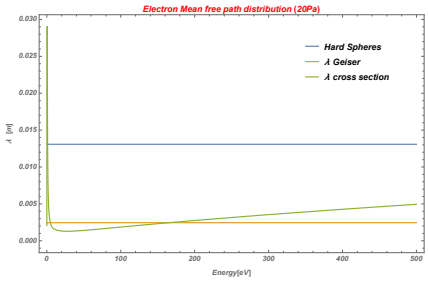
(a) *Collision time distribution in function of the neutral pressure*

(b) *Mean free path distribution in function of the neutral pressure*

Fig. 2.2: *In figure a) we show the trend of the collision time in function of the neutral pressure, while in figure b) the trend of the mean free path, always in function of the neutral pressure, for the various models presented in section 2.3. Electrons are the "bullets" and the Argon neutral particles are the "target" species. For electrons we have considered a fixed energy of $15eV$ that is the thermal agitation energy for electrons at temperature of $116000K$. It is evident that, as the neutral pressure increases, the mean free path and the collision time between two successive collisions decreases. The presence of an electric field moves toward the curve plotted above reducing the time-step and increasing the computation cost. In Figure a) we have also indicated with a straight line the simulation time that, in this configuration, is of $1ns$.*



*(a)*Collision time distribution in function of the electron energy

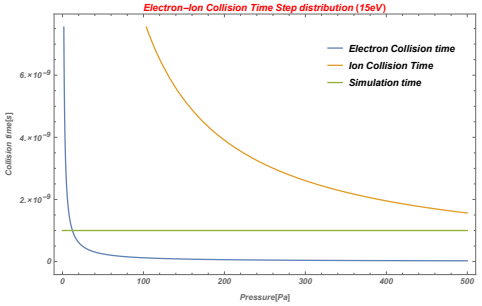


*(b)*Mean free path distribution in function of the electron energy

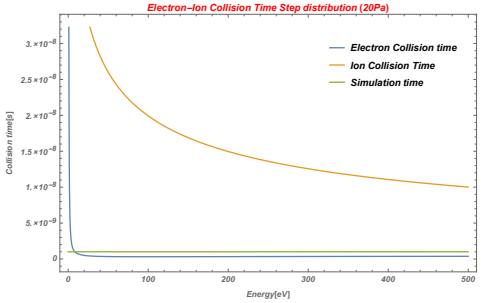
Fig. 2.3: *In figure a) we show the trend of the collision time in function of the electron energy (computed in eV); while in figure b) the trend of the mean free path, always in function of the electron energy, for the various models presented in section 2.3. Electrons are the "bullets" and the Argon neutral particles are the "target" species. We have considered a pressure for the neutral background species of 20Pa. It is evident that, as the energy increases, the collision time between two successive collisions decreases. The presence of a greater pressure moves toward the curve plotted above, reducing the time-step and increasing the computation cost. In Figure a) we have also indicated with a straight line the simulation time, that in this configuration is of* $1ns$.

We conclude this subsection noting that in all the proposed figures, we have tacitly considered only the cross sections of collisional processes involving electrons. The reason of this is shown in fig. 2.4.

The electrons have much higher speed of the ions due to their lower mass. For this reason, the characteristic collision time of electron collisions is much lower if compared with the ion collision time. From this follows that, to choose the global time step for a particular simulation, it is only necessary to choose a time-step compatible with the particular studied electron configuration. In this way, as shown in fig. 2.4, this time step is also suitable to study the ion motion.



*(a)*Collision time distribution in function of the neutral pressure



*(b)*Collision time distribution in function of the energy of the incident particle

Fig. 2.4: *In figure a) we show the trend of the collision time in function of the neutral pressure, while in figure b) the trend of the same quantity in function of the energy of the incident particle. In figure a) we have kept fixed the energy at* $15eV$*, while in figure b) we have kept fixed neutral pressure at* $20Pa$*. It is clear that, for the same energy or pressure, the characteristic time between two collisions involving an electron is much lower if compared with the collision time of an ion. In Figure a) and b) we have also indicated with a straight line the simulation time, that in this configuration is of* $1ns$.

2.3.2 A new algorithm to take into account neutral pressure variation

Starting from the discussions of the previous section, we have decided to modify Vahedy and Surendra's scheme in order to untie the global time step Δt from the background density of the target specie n_t. In our formulation, for each particle, at the beginning of each time-step, using equations 2.23, the collision time τ_i is calculated.
In the following we will denote with τ_{1i} the first calculation of the collision time τ_i for the particle i, τ_{2i} the second calculation of the collision time τ_i for the particle i, and so on.
If the collision time τ_{1i} is less than the global time-step Δt

$$\tau_{1i} \leq \Delta t \tag{2.24}$$

the particle, after advancing with a suitable integrator by a time equal to τ_{1i}, is made to collide, and the type of collision is checked in agreement with 2.8.
Immediately after the collision, the new time interval τ_{2i} is calculated taking also into account the new particle velocity.
If $(\tau_{1i} + \tau_{2i}) \leq \Delta t$, after a new particle advance, a new collision occurs.
The cycle stops when at the N-th step (this quantity is computed at the $(N-1)$-th step)

$$\left(\sum_{j=1}^{N} \tau_{ji} - \Delta t \right) > 0 \tag{2.25}$$

If then holds

$$\left(\Delta t - \sum_{j=1}^{N-1} \tau_{ji} \right) = 0 \tag{2.26}$$

the previous cycle is repeated for another particle; otherwise the particle i is advanced for a time equal to

$$\Delta t - \sum_{j=1}^{N-1} \tau_{ji} \tag{2.27}$$

and we check if a collision occurs testing the following condition:

$$r \leq \frac{\left(\Delta t - \sum_{j=1}^{N-1} \tau_{ji} \right)}{\tau_{Ni}} \tag{2.28}$$

where r is a random number with a uniform distribution between zero and one.
The cycle presented is performed only if, at the beginning, the inequality 2.24 is true.
However, this condition, in a high-vacuum regime, is not strictly fulfilled. In this

case we advance particle for a time equal to Δt and then we check if a collision occurs testing the following condition

$$r \leq \frac{\Delta t}{\tau_{1i}} \tag{2.29}$$

In this way, using 2.29, we are also able to simulate in a consistent way collisions at low pressure.
In pseudocode 1 we underline the main parts of the new algorithm; to lighten the notation we have neglected the parts in witch we stop the execution and we change particle if the particle exits from the domain.

Comparison with Vahedy and Surendra's scheme in low pressure condition

Took note of the algorithm showed in pseudocode 1; in this subsection we want to show that equations 2.28,2.29, are fully compatible with Vahedy and Surendra's scheme.
In fact the request 2.10 is nothing but the equation 2.29, using mean free paths defined in equations 2.19, 2.20.
Performing calculations in 2.10 we can rewrite it as:

$$\Delta s_i \sigma_T(E_i) n_t \leq 0.1 \longrightarrow \frac{\Delta s_i}{\lambda_i} \leq 0.1 \tag{2.30}$$

In a similar way equation 2.29 can be rewritten as

$$r \leq \frac{\Delta t}{\tau_{1i}} \longrightarrow r \leq \frac{|\mathbf{v_i}|\Delta t}{\lambda_i} = \frac{\Delta s_i}{\lambda_i} \tag{2.31}$$

using 2.22 for τ_{1i} and using the fact that $\Delta s_i = |\mathbf{v_i}|\Delta t$.
In such a way, in our model, equation 2.10 acquires meaning, in fact, compared with 2.29, it requires to have a probability of collision in a time-step less or equal to 10%. In other words, condition 2.10 requests to be in a low pressure condition i.e. the particle can not reach out multiple collision events in the same global time-step. However, as already stated, unlike Vahedy and Surendra's scheme, we are not obliged to force the choice of the time-step to fulfill condition 2.10.

Algorithm 1 MonteCarlo collision model

procedure MONTECARLO COLLISION
  for species s $\leq N_{species}$ **do**
    for particle i $\leq N_{particle}$ **do**
      l=1
      Compute τ_{lis} with 2.23
      while $\sum_{k=1}^{l} \tau_{kis} \leq \Delta t$ **do**
        while-iteration=TRUE
        Particle advance for τ_{lis} and collision with 2.8
        Particle Tracking
        l+=1
        Compute new τ_{lis} with 2.23
        if $\sum_{k=1}^{l} \tau_{kis} > \Delta t$ **then**
          if $\left(\Delta t - \sum_{k=1}^{l-1} \tau_{kis}\right) > 0$ **then**
            Particle advance for $\Delta t - \sum_{k=1}^{l-1} \tau_{kis}$
            Particle Tracking
            if $r \leq \frac{\left(\Delta t - \sum_{k=1}^{l-1} \tau_{kis}\right)}{\tau_{lis}}$ **then**
              Particle collision with 2.8
            end if
            Break
          else
            if $\left(\Delta t - \sum_{k=1}^{l-1} \tau_{kis}\right) == 0$ **then**
              Break
            end if
          end if
        end if
      end while
      if while-iteration==FALSE **then**
        Particle advance for Δt
        Particle Tracking
        if $r \leq \frac{\Delta t}{\tau_{lis}}$ **then**
          Particle collision with 2.8
        end if
      end if
    end for
  end for
end procedure

2.3.3 Recombination Process

In the previous section we have shown the main characteristics of our algorithm. In this section we want to extend the previous formulation to include also the recombination process between a charged electron and a charged ion.

We will propose an approach similar to that one proposed in [2]. However unlike [2], our algorithm is fully integrated with standard PIC formulation and therefore

the recombination process becomes competitive with other processes described in previous sections.
To explain our recombination formulation we will proceed by steps. The first step is to show the recombination cross section.
In our algorithm we have used Kramers's formula defined by:

$$\sigma_{\text{recombination}}(n, E_{elec}) = 2.105 \times 10^{-22} \frac{{R_y}^2 Z^4}{nE_{elec}(n^2 E_{elec} + R_y Z^2)} \text{ cm}^2 \tag{2.32}$$

where E_{elec} is the energy of the chosen electron, R_y is the Rydberg constant and n is the principal quantum number of the recombined ion. See [1], [24] to have more details. The Kramers's formula can also be used to calculate recombination cross section for non-bare ion by introducing an appropriate charge, called effective charge Z_{eff}.
To estimate it, a simple expression was given in [42] or [41]

$$Z_{eff} = \frac{1}{2}(Z_c + Z_I) \quad \text{for} \quad Z_C \geq Z_I \geq \frac{Z_C}{2} \tag{2.33}$$

and

$$Z_{eff} = \sqrt{Z_C Z_I} \quad \text{for} \quad \frac{Z_C}{2} \geq Z_I \geq 1 \tag{2.34}$$

where Z_C is the nuclear core charge and Z_I is the ionic charge before electron capture.

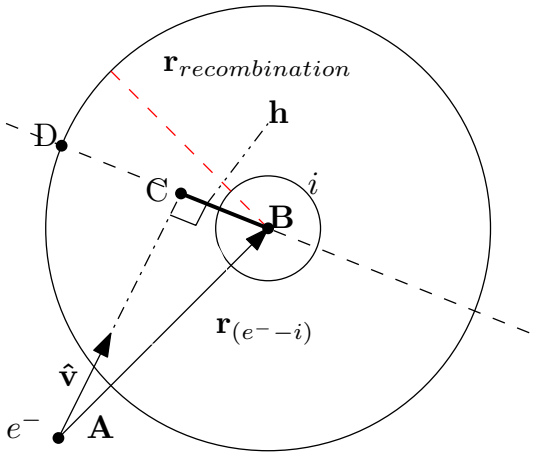


Fig. 2.5: *Schematic figure to show the model implemented for electron-ion recombination.*

Took note of this, in the following we will show the algorithm implemented.
Fixed an electron and a possible ion with which the electron recombines, the first step is to change the reference system in which we analyze the collision, in particular it is necessary to study the recombination in the reference system in which the ion is fixed.

Then using equation 2.32, we compute the radiative recombination cross section and we define a "recombination sphere" of radius $r_{\text{recombination}}$ in the following way

$$r_{\text{recombination}} = \sqrt{\frac{\sigma_{\text{recombination}}}{\pi}} \tag{2.35}$$

If an electron impacts in this sphere, the recombination occurs and a new neutral particle is created.
Precisely, looking at figure 2.5, and calling as $\mathbf{B}$ and $\mathbf{A}$ the ion and electron positions in the chosen reference system, it is necessary to compute the relative position of the ion respect to the electron in the following way

$$\mathbf{r}_{(\mathbf{e^- - i})} \equiv \mathbf{B} - \mathbf{A} \tag{2.36}$$

Then, thanks to 2.36, and using the versor of the electron velocity defined in figure 2.5 as $\hat{\mathbf{v}}$; a recombination process could occur if the following inequality is verified[3]

$$\mathbf{r}_{(\mathbf{e^- - i})} \cdot \hat{\mathbf{v}} > 0 \tag{2.37}$$

If 2.37 holds, the following quantity is computed

$$h = \sqrt{(|\mathbf{r}_{(\mathbf{e^- - i})}|)^2 - (\mathbf{r}_{(\mathbf{e^- - i})} \cdot \hat{\mathbf{v}})^2} \tag{2.38}$$

and, finally, if results that $h \leq r_{\text{recombination}}$ or $|\mathbf{r}_{(\mathbf{e^- - i})}| \leq r_{\text{recombination}}$ the recombination process takes place.
The previous cycle is proposed in the following pseudocode

Algorithm 2 Recombination Model

```
1:  procedure RECOMBINATION MODEL
2:      Fixed an electron with velocity versor v̂
3:      for ions ∈ "competitive mesh entities" do
4:          Change reference system
5:          Compute r_recombination using 2.35
6:          Compute r_(e- - i) using 2.36
7:          if r_(e- - i) · v̂ > 0 then
8:              Compute h = sqrt((|r_(e- - i)|)^2 − (r_(e- - i) · v̂)^2)
9:              if (h ≤ r_recombination || r_(e- - i) ≤ r_recombination) then
10:                 Recomb-Performed=TRUE
11:                 Break
12:             end if
13:         end if
14:     end for
15: end procedure
```

[3] " · " is the usual dot product

As shown in the previous pseudocode, fixed an electron, it is not necessary to compute $r_{\text{recombination}}$ for all the ions presented in the domain, but only for those which belong to the so called "competitive mesh entities".
With this term we refer to the elements of the mesh that are competitive to the mesh element that contains the fixed electron. In other words, if the mesh entity is a tetrahedron (like in F3MPIC) it is sufficient to test only the ions that belong to the tetrahedron in which there is the chosen electron and for ones that belong to the tetrahedra adjacent to it.
This fact is closely connected with the condition of Courant-Friedrichs-Lewy [33], in fact to avoid aliasing errors, it is necessary that, chosen a time-step, the average displacement of a particle, within the same global time step, is not greater of the characteristic mesh size.
We conclude this section noting that the recombination process begins to be competitive with other processes proposed in section 2.3 only at very high density.
The reason is strictly connected with recombination cross section. In fact this one is several orders of magnitude smaller than the one calculated previously thus making the amount $r_{\text{recombination}}$ extremely small.
For this reason, only at very high density, it is possible to have a considerable fraction of recombinant particles.
We conclude this section integrating pseudocode 2 with the previous one. The result is proposed in pseudocode 3, where we have underlined the new recombination phase.

Algorithm 3 MonteCarlo collision+Recombination

```
1: procedure MonteCarlo collision
2:     for species s ≤ N_species do
3:         for particle i ≤ N_particle do
4:             l=1
5:             Compute τ_lis with 2.23
6:             if ( s ∈ electron) then
7:                 Test Recombination with pseudocode 2
8:                 if Recomb-Performed==TRUE then
9:                     Recomb-Cycle=TRUE
10:                    Continue
11:                end if
12:            end if
13:            while Σ_{k=1}^{l} τ_kis ≤ Δt do
14:                while-iteration=TRUE
15:                Particle advance for τ_lis and collision with 2.8
16:                Particle Tracking
17:                l+=1
18:                Compute new τ_lis with 2.23
19:                if ( s ∈ electron) then
20:                    Test Recombination with pseudocode 2
21:                    if Recomb-Performed=TRUE then
22:                        Recomb-Cycle=TRUE
23:                        Break
24:                    end if
25:                end if
26:                if Σ_{k=1}^{l} τ_kis > Δt then
27:                    if (Δt − Σ_{k=1}^{l−1} τ_kis) > 0 then
28:                        Particle advance for Δt − Σ_{k=1}^{l−1} τ_kis
29:                        Particle Tracking
30:                        if r ≤ (Δt−Σ_{k=1}^{l−1} τ_kis) / τ_lis then
31:                            Particle collision with 2.8
32:                        end if
33:                        Break
34:                    else
35:                        if (Δt − Σ_{k=1}^{l−1} τ_kis) == 0 then
36:                            Break
37:                        end if
38:                    end if
39:                end if
40:            end while
41:            if while-iteration==FALSE AND Recomb-Cycle==FALSE then
42:                Particle advance for Δt
43:                Particle Tracking
44:                if r ≤ Δt/τ_lis then
45:                    Particle collision with 2.8
46:                end if
47:            end if
48:        end for
49:    end for
50: end procedure
```

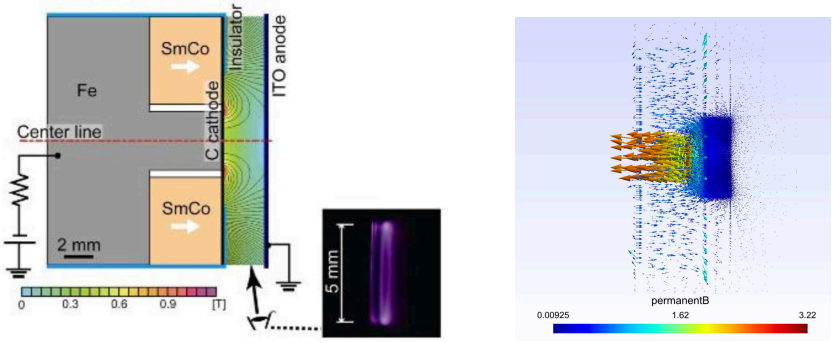
2.4 Main approximations and simulation results

In the previous section we have shown the main cycle of our algorithm schematically presented in pseudocode 1.
In this section we want to show the main approximations that we have implicitly done in the previous sections and we want to validate the new algorithm with some simulation results.
As it has been shown in presudocode 1, in the same global time step Δt and in a suitable pressure condition, particles position is updated many times using an integrator which interpolate fields in particles position.
This clearly shows that the fields updating is done only at any Δt. This feature is innovative if compared with other PIC codes where fields updating is done after each particles movement.
In line of principle the decision to update fields only at each Δt could bring substantial errors in the integration phase, in fact the charge distribution changes several times within the same global time step. The main advantage of this approximation is to reduce computational cost, in fact having untied the global collision time step Δt from the density of neutral particles, we are not obliged to choose a time step of the same order of magnitude of τ, and, thanks to this, it is not necessary to recalculate at each τ a range of quantities (such as electric and magnetic fields) that are typically computationally bundersome.
In this way, paying a slight increase of calculations during each time-step, we are able to reduce by several orders of magnitude the overall number of global time steps to achieve the same global simulation time of other PIC codes.
To validate this first assumption it is necessary to ensure that the fields variation due to particles motion during the same global time-step, is negligible if compared with the same variation calculated at any global time-step. In other words it is necessary to ensure that the percentage variation between the fields computed at each global time-step Δt, and the field calculated at each τ, is small and generally compatible with statistical fluctuations.
To characterize this percentage fluctuation we will indicate with $\mathbf{E}_{\Delta t}$ the electric field calculated at each global time-step, and with $\mathbf{E}_{\tau}$ the electric field computed at each collisional time step τ (As stated, in Vahedy and Surendra's model $\tau \sim \Delta t$ so there is no need to make this difference.)[4]. Using this notation and assuming to be $\mathbf{E}_{\tau}$ the quantity with the least error, the percentage variation introduced above can be characterized with a piecewise function defined as

$$\text{P.Variation}_{ij}[\%] = \left(\frac{|\mathbf{E}_{\Delta t_i}| - |\mathbf{E}_{\tau_j}|}{|\mathbf{E}_{\tau_j}|}\right) \times 100 \qquad \forall i \in [1, N_{\Delta t}] \quad \forall j \in [1, N_{\tau}] \tag{2.39}$$

[4]To obtain this quantity we have done a simulation choosing $\Delta t \equiv \tau$

with $N_{\Delta t}$ number of global time-steps Δt and N_τ number of "collision" time steps. The error on this quantity follows from propagation.
The electric field is a punctual quantity, for this reason in line of principle it is possible to define a function of this type in each point of the domain of interest, for example a node of the mesh.



(a)General scheme of the apparatus used to test the new code

(b)Magnetic field distributions (in a full 3D configuration) reproduced in F3MPIC

Fig. 2.6: *Left: In figure a) we have reproduced a scheme of the simulated apparatus. A magnetized DC micro-discharge has formed the basis of a micro-propulsion thruster as described in [20],[21]. Right: The magnetic field configuration reproduced in F3MPIC in a full 3D configuration. The magnetic circuit is located behind the cathode.*

Let's show now the system that we have used to compute the percentage variation just defined.
We have reproduced a magnetized DC micro-discharge as shown in figure 2.6. These discharges have formed the basis of a micro-propulsion thruster (see [20]) that can be used as a plasma contactor or neutralizer.
The simulated plasma discharge develops between two 5 mm-diameter circular electrodes spaced 3 mm apart.
The potential is set to 200V and it is made between an anode and a cathode that are the two basics of the micro-propulsion thruster. This discharge was experimentally investigated by Ito (see [21]); its geometry is reproduced in the simulation and the 3D particles dynamic is studied in detail. The electrostatic field is solved either on a 2D longitudinal plane with the assumption of axisymmetry or in full 3D without restricting assumptions.
In figure 2.7 is shown the configuration of the electric field.

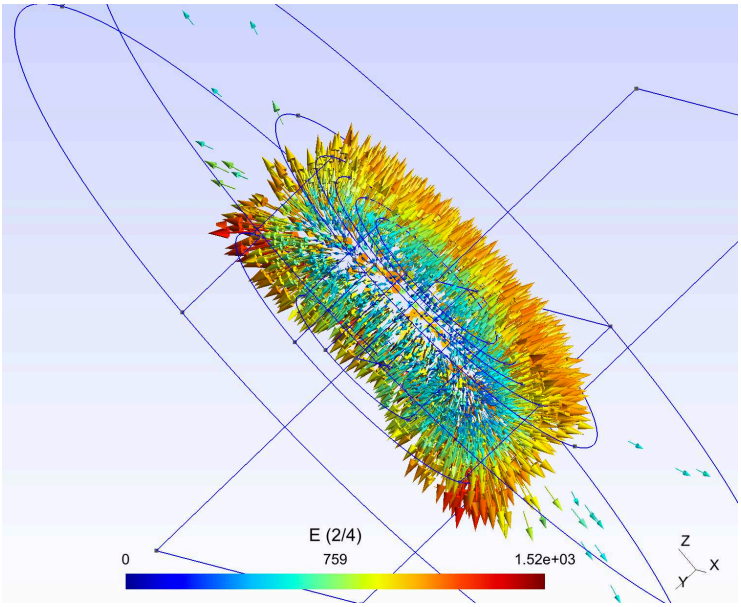


Fig. 2.7: *Electric field configuration reproduced in F3MPIC with a system at convergence. The plasma bulk is shielded*

We have analyzed two different configurations of the plasma thruster presented above.

- In the first one we have analyzed only the electric fields (i.e. we have switched off the magnetic fields) to appreciate electric fields variation

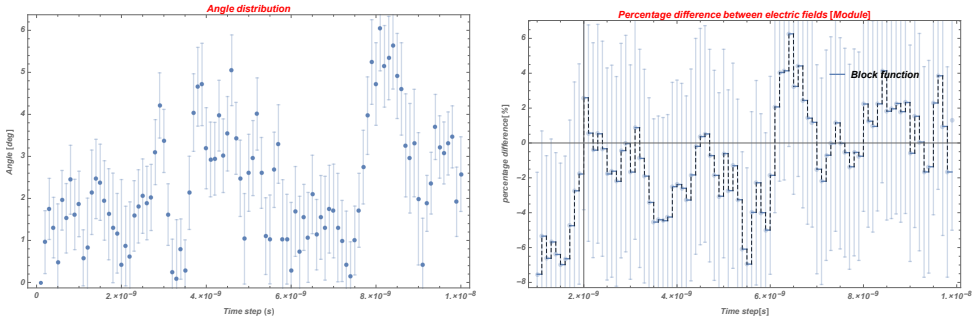
- In the second one we have also reproduced the magnetic field. Experimentally the magnetic field is obtained thanks to a ring-shaped axially-magnetized permanent magnet and an iron core for field shaping. The magnetic circuit is located behind the cathode.

In both cases, in the system there was a uniform background of neutral particles having a density that was function of the chosen pressure.
In what follows we have started simulations with a prefixed number of electrons and ions in the system (loading them in a "source" region) and we have observed the variation of them. A constant-rate source inserts at each time step a fixed number of ions and electrons in the system. These ones are inserted in the system with a kinetic energy compatible with their thermal kinetic energy (for ions we have selected a temperature of $300K$ while for electrons of $116000K$).
Starting with only electric fields we have characterized electric fields fluctuations in term of P.Variation$_{ij}[\%]$. To avoid aliasing errors in the calculation of the electric fields, we have chosen 240 near nodes of tetrahedral mesh in plasma bulk with a system to convergence, i.e. after the initial transient, with a mean ion density of $10^{13}m^{-3}$, and we have calculated the mean electric fields defined as $< \mathbf{E}_{\Delta t_i} > \equiv$

$(< E_{x\Delta t_i} >, < E_{y\Delta t_i} >, < E_{z\Delta t_i} >)$ (at each Δt_i) and $< \mathbf{E}_{\tau_j} > \equiv (< E_{x\tau_j} >, < E_{y\tau_j} >< E_{z\tau_j} >)$ (at each τ_j) on these nodes. In this way, using 2.39, we are able to characterize the changing of the magnitude of the electric field.

To characterize also its variation in direction, we have introduced a new quantity called θ. This one is the angle between the electric field vectors $< \mathbf{E}_{\Delta t_i} >$ and $< \mathbf{E}_{\tau_j} >$ and it is computed taking these two vectors as coplanar.

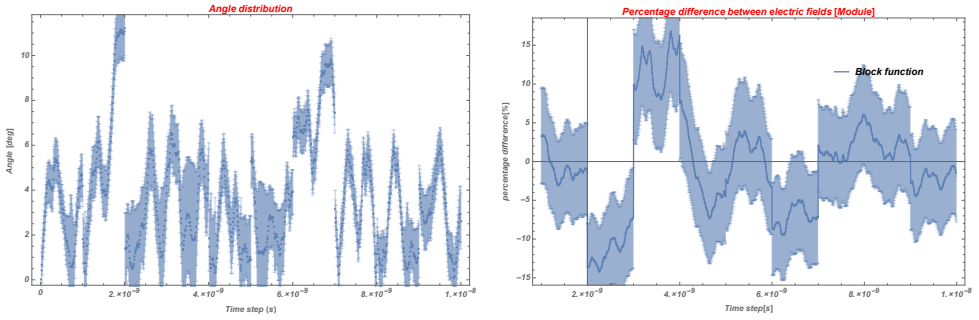
In what follows, with a system at convergence, we have analyzed 10 global time-steps Δt ($N_{\Delta t} = 10$) each lasting $1ns$, τ was chosen in accordance with neutral pressure. Figure 2.8 and 2.9 show the results obtained. In the first one we have chosen a neutral pressure of 20 Pa and in the second one a neutral pressure of 200 Pa. The former, in agreement with 2.10 has $\tau_j \sim 10^{-10}$; the latter $\tau_j \sim 10^{-11}$.



(a)Angle analysis

(b)Percentage variation of the module of the electric field

Fig. 2.8: *Left: variation of the angle between the vectors $< \mathbf{E}_{\Delta t_i} >$ and $< \mathbf{E}_{\tau_j} >$ calculated at each τ_j. In this particular configuration, the neutral pressure is fixed at 20 Pa with $\tau_j \sim 10^{-10} \quad \forall j \in [1, 100]$ Right: Percentage variation of the module of the electric field calculated using P.Variation$_{ij}[\%]$ defined in 2.39. The error bars follow from error propagation.*



(a)Angle analysis

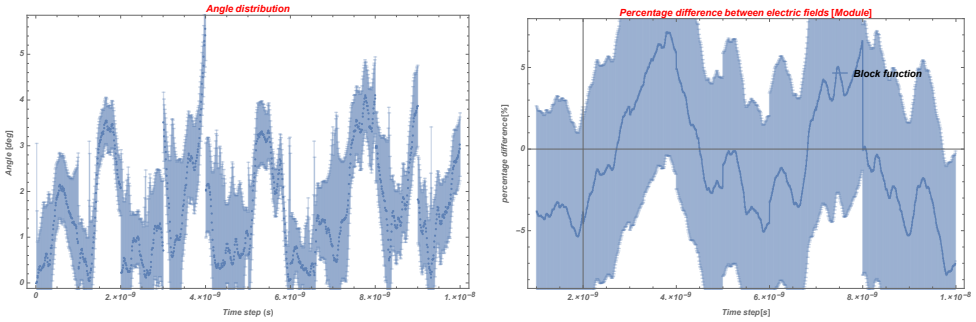
(b)Percentage variation of the module of the electric field

Fig. 2.9: *Left: variation of the angle between the vectors $< \mathbf{E}_{\Delta t_i} >$ and $< \mathbf{E}_{\tau_j} >$ calculated at each τ_j. In this particular configuration, the neutral pressure is fixed at 200 Pa with $\tau_j \sim 10^{-11} \quad \forall j \in [1, 1000]$ Right: Percentage variation of the module of the electric field calculated using P.Variation$_{ij}[\%]$ defined in 2.39. The error bars follow from error propagation*

As shown in figure 2.8,2.9, there is no evidence of an exponential growth error due to the increase of the pressure and the "error" supposed linked to $\Delta t \not\sim \tau$ and summarized by P.Variation$_{ij}[\%]$, is very small. The results show that the change in direction between the vectors$< \mathbf{E}_{\Delta t_i} >$ and $< \mathbf{E}_{\tau_j} >$ calculated at each τ_j is at most about 10 degrees, while the percentage variation in the module of the electric field presents discard on average of 5% degree. The big errors on these quantities are compatible with statistical fluctuations and the plot itself is statical, in the sense that it is bound to the particular charge configuration which is reached in the case plotted; analog simulations show a similar behavior.

Finally it is interesting to note that, as shown in figure 2.9, there are some "misleading spikes". These spikes are mainly related to the p_{2c}, in fact, in PIC standard, we use computational particles. In this particular implementation each computational particle matches to 10^2 real particles. When particles move and cross mesh boundaries we observe "spikes" related to the "step" changes of the field due to the "step" motion of particles. To validate this hypothesis we have done the same analysis of the one propose in figure 2.9 but, in this case, we have lowered p_{2c} of an order of magnitude. We have also increased the source rate and the number of the initial loaded particles in the system, at the purpose to have the same number of overall particles of figure 2.9, for both charged species. The results are proposed in figure 2.10.

The "noise" is greatly reduced and the overall errors, in both the magnitude of electric field and the angle, are significantly reduced. These fluctuations could be further reduced increasing also the density of the charged particles presented in the system improving, in this way, the convergence achieved.

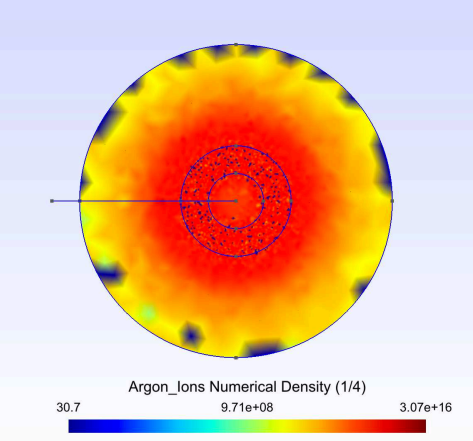


(a)Angle analysis

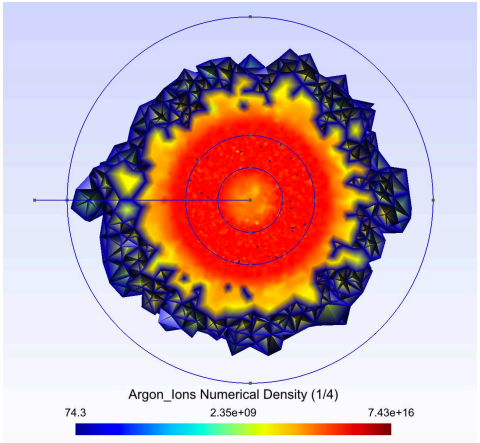
(b)Percentage variation of the module of the electric field

Fig. 2.10: *Left: variation of the angle between the vectors $< \mathbf{E}_{\Delta t_i} >$ and $< \mathbf{E}_{\tau_j} >$ calculated at each τ_j. In this particular configuration the neutral pressure is fixed at 200 Pa, with $\tau_j \sim 10^{-11} \quad \forall j \in [1, 1000]$. Right: Percentage variation of the module of the electric field calculated using P.Variation$_{ij}[\%]$ defined in 2.39. The error bars follow from error propagation. If compared with figure 2.9, we have lowered p_{2c} of an order of magnitude. We have also increased the source rate and the number of loaded particles at the purpose to have the same number of overall particles of figure 2.9*

The good results obtained ensure that the percentage variation between the fields updating at each global time-step Δt, and the field updating computed at each τ_j, is small validating in this way our model for a convergence system.
Things get worse in a system not at convergence i.e. during the initial transient.[5]
However, in this case the same $< \mathbf{E}_{\tau_j} >$, that would be used for the integration phase during the collision time step τ_j, could be seen as an approximation and it is not suitable to describe without too much errors the advancement of the system. In fact, at the beginning, the system is devoid of charged particles and this leads to large variations of fields also for small movements of the charged particles.
Generally the initial phase is a problem that is shared by all PIC codes and it is difficult to eliminate, our model is no exception. In line of principle it would be necessary to reduce the time step as much as possible i.e. $\Delta t \equiv \tau \to 0$ but, for obvious reasons, it is not possible.
Until now we have only validated the first fields assumption using only electric fields; in what follows we will apply our algorithm to analyze the behavior of the system in figure 2.6, in function of neutral pressure.
We will use both magnetic and electric fields as shown in figure 2.7 and 2.6b).



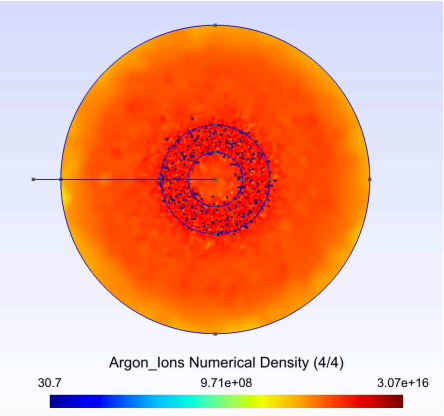
(a) *Argon Ions density with a full 3D configuration of fields, after 1000 time-steps. Neutral pressure:* 20*Pa*



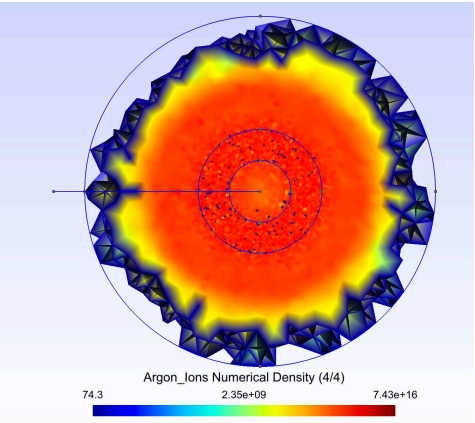
(b) *Argon Ions density with a full 3D configuration of fields, after 1000 time-steps. Neutral pressure:* 200*Pa*

Fig. 2.11: *Left: Numerical argon Ions density after 1000 time-steps. The neutral pressure is of* 20*Pa. Right: Numerical argon Ions density after 1000 time-steps. The neutral pressure is of* 200*Pa. In both figures the scale is logarithmic and we started the simulation with a fixed number of particles in the system. A constant-rate source inserts at each time step a fixed number of ions and electrons in the system*

[5] In principle also the walls might be a problem. However in general the walls can be treated as a delayed "bulk"; once the global convergence is achieved the results start to be consistent



(a)Argon Ions density with a full 3D configuration of fields, after 4000 time-steps. Neutral pressure: 20P



(b)Argon Ions density with a full 3D configuration of fields, after 4000 time-steps. Neutral pressure: 200P

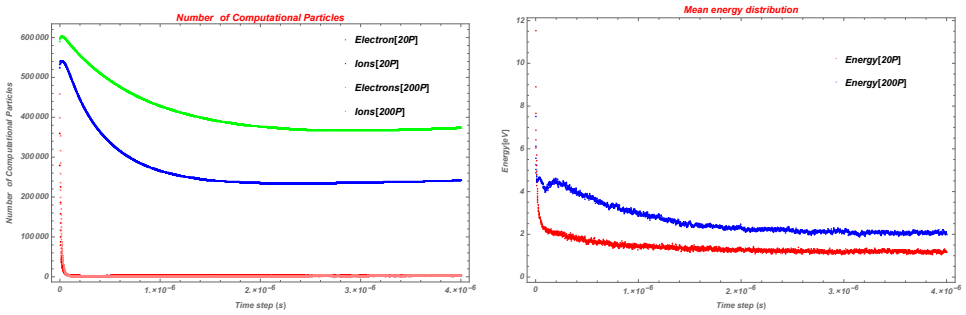
Fig. 2.12: *Left: Numerical argon Ions density after 4000 time-steps. The neutral pressure is of 20Pa. Right: Numerical argon Ions density after 4000 time-steps. The neutral pressure is of 200Pa. In both figures the scale is logarithmic and we started the simulation with a fixed number of particles in the system. A constant-rate source inserts at each time step a fixed number of ions and electrons in the system.*

In figure 2.11 and 2.12 we have shown the results obtained for the same system but with different neutral pressure. In both cases we have plotted the argon ions numerical density respectively after 1000 and 4000 time steps. In figures 2.11a) and 2.12a), the neutral pressure is of 20Pa while in figures 2.11b) and 2.12b), the neutral pressure is of 200Pa.

In figures 2.11b) and 2.12b) it is clear as the system evolves more slowly due to the higher pressure and due to the mayor number of interactions at which the particles are subjected.

We conclude noting that we have preferred to analyze the numerical density of the argon ions in the system but, the same analysis, can be done looking at the number of computational particles in the system. See figure 2.13a).

In this case we have plotted the number of computational electrons and ions in function of the simulation time. As stated, the number of these ones changes in different way as function of the neutral pressure. In figure 2.13b), we have reported the average kinetic energy of the electrons in the system.



(a) *Number of computational particles as function of the simulation time*

(b) *Mean electron energy distribution as function of the simulation time*

Fig. 2.13: *Left: Number of computational particles in presence of magnetic field and electric field as function of the simulation time. As evident the number of ions and electrons is significantly higher when in the system there is greater pressure. We also note that the number of electrons decreases faster due to higher electron speed. Right: Mean electron energy distribution as function of the simulation time [the energy is measured in eV]. In this figure we avoid, for clarity, to add error bars on mean energies.*

Results obtained in figure 2.13b) are also connected with another important approximation that we have done in the previous sections. This approximation is linked with equation 2.8.

As stated, to choose the type of collision it is necessary to test condition 2.8 calculating the ratio between collision frequencies of all process and 2.6 that, in principle, could change during the simulation and, for this reason, it should be necessary to recompute it before to test equation 2.8.

However, as shown in figure 2.13b), in a convergence system the percentage variation of this quantity (i.e. max electron energy) is very low, therefore it is sufficient to compute it at the beginning of the convergence phase and to test only ratio 2.8 to choose the type of collision without recompute 2.6.

The overall obtained results ensure the goodness of model proposed; to conclude this section we want to underline that, another advantage of our algorithm compared with other ones, is that it can be easily extended also in a non-uniform gas density distribution. In fact, we do not choose random which particle collide, but we only choose random the collision type in accordance with 2.8.

In such a way our algorithm can also be applied in a non-uniform pressure condition but, in this case, it is necessary to compute the density of the background species n_t, looking at the neutral particles presented in the tetrahedron (i.e. mesh entity) in which the bullet is, before to use 2.19 .

2.5 Conclusions

A new Monte Carlo collision model has been developed. The new code is fully compatible with PIC standard, in particular we have shown that our formulation is valid for arbitrary neutral pressure and at low pressure value is compatible with V. Vahedi, M. Surendra's model.
A new type of recombination model has also been introduced and made compatible with the new Monte Carlo model developed. Simulation results show a consistent behavior of the system when pressure changes. The approximations made in the construction of the new algorithm have been extensively tested, using as a test geometry, a micro-propulsion thruster described in [20],[21].
In particular, the obtained results show that these approximation introduce acceptable errors and do not affect the overall goodness of the new model. The new code, compared with other competitive Monte Carlo collision models is also more efficient and generally a faster algorithm. We conclude noting that the new presented algorithm is also compatible with arbitrary grid and arbitrary tracking method. This new Monte Carlo algorithm is now fully implemented in F3MPIC, a three dimensional PIC code of Padua University.

3 Implementation of a new charge conserving method

> *Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level.*
>
> — **Enrico Fermi**

Abstract

There are several numerical techniques to solve the continuity equation locally, which allow us to avoid solving Poisson's equation at every time step.

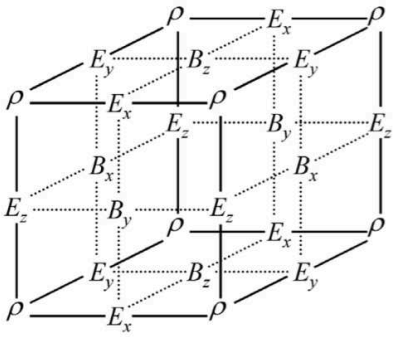


Fig. 3.1: *Example of structured mesh with computation of charge density and the electromagnetic fields*

They are called "charge conservation methods". As described by Eastwood [40], a charge flux of a particle can be computed from the start and end points of the particle movement when both start and end points are located in the same cell. When a particle moves across the cell meshes, the particle movement is assigned to separate motions in each cell by the cell meshes. From superposition of charge flux, charge conservation can be realized for any particle trajectories made up of straight line segments between any start and end points.

In charge conservation methods, a particle trajectory over one time step is conventionally assumed to be a straight line. However, when a particle moves from (x, y, z) to $(x + v_x t, y + v_y t, z + v_z t)$ for one time step, the particle can take an arbitrary

trajectory as long as it does not move more than one grid spacing. In other words, a particle trajectory needs not to be a straight line. Recently, a new charge conservation method was developed by Umeda (see [37],[38]) assuming that a particle trajectory is a zigzag line. In this chapter we have developed a charge-conserving algorithm following Umeda's article and we have adapted it to F3MPIC that, as anticipated, uses an unstructured mesh. The original algorithm of Umeda is, in fact, based on a structured cubic mesh. The basic idea will be to combine two meshes: a cubic one, in which the current density deposition is made, and a structured one, in which the field integration phase and the subsequent advancement of the particles are managed. See figure 3.1

3.1 Introduction

Numerical simulations of plasmas of charged particle beams are modeled using the Vlasov-Maxwell system and are often performed using the Particle-In-Cell method. If we consider a non collisional plasma constituted of charged particles, the evolution of these particles is described by a distribution function $f(\mathbf{x}, \mathbf{u}, t)$ depending on space $\mathbf{x} \in \mathbb{R}^3$, momentum $\mathbf{u} \in \mathbb{R}^3$, and time $t > 0$, which satisfies the following equation called "Vlasov's equation"

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q(\mathbf{E} + \mathbf{v} \times \mathbf{B})}{m} \cdot \nabla_{\mathbf{u}} f = 0 \tag{3.1}$$

The electromagnetic field $(\mathbf{E}, \mathbf{B})$ is described by the Maxwell's equations and the coupling with the Vlasov's equation is done through the source terms $(\rho, \mathbf{J})$ such that

$$\frac{d\mathbf{E}}{dt} = c^2 \mathbf{rot}\ \mathbf{B} - \frac{\mathbf{J}}{\epsilon_0} \quad \frac{d\mathbf{B}}{dt} = -\mathbf{rot}\ \mathbf{E} \tag{3.2}$$

$$\text{div}\ \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \text{div}\ \mathbf{B} = 0 \tag{3.3}$$

By integrating the Vlasov's equation for all $\mathbf{u} \in \mathbb{R}^3$ we get

$$\frac{d\rho}{dt} + \text{div}\ \mathbf{J} = 0 \tag{3.4}$$

The last relation, called continuity equation, is crucial. Indeed, it ensures that the Vlasov-Maxwell's system (3.3+3.2+3.1) is well-posed.
It can be shown ([31]) that relations 3.3 are verified at any time, as soon as they are initially satisfied, and the sources terms $\rho, \mathbf{J}$, satisfy the continuity equation.
In this way remains only to solve evolution equations 3.2. The previous property represents a great advantage in numerical simulations because Vlasov-Maxwell's system can then be implemented without solving equations 3.3 and especially without

solving the Poisson's equation for the electric potential.
Unfortunately, classical PIC codes that use the so called Cloud-In-Cell algorithm, do not satisfy the continuity equation locally and, as a consequence, errors may appear in the irrotational part of the electric field in Gauss's law.[1]
Therefore electromagnetic PIC solvers need to deal with this issue, either by performing a field correction or by computing the current density in a specific way, so as to enforce the validity of the continuity equation.
There are several numerical techniques for solving the continuity equation locally, which allow us to avoid solving Poisson's equation at every time step. They are called "charge conservation methods". In the Villasenor-Buneman method [22], a particle trajectory over one time step is assumed to be a straight line. Therefore all start and end points of the particle trajectory segments are located along the straight line. The method proposed by Villasenor-Buneman is, unfortunately, very complex, because it is necessary to compute the intersection points between the particle trajectory and cell meshes. These computations are realized with several "IF" statements not so easy to implement.
Recently a new charge conserving method was developed by Umeda [37] assuming that a particle trajectory is a zigzag line. The algorithm of the zigzag scheme is realized without "IF" statements. The main advantage of this technique, compared to the other ones, is not to modify the electromagnetic field away from the source which may generate errors for some applications. The only disadvantage of this scheme is connected with the fact that this formulation is strongly linked to the use of a Yee's solver on a regular grid to solve the continuity equation. In F3MPIC we use an unstructured mesh of tetrahedra and, for this reason, Umeda's algorithm is not directly applicable.
Recently a new scheme which achieves "charge conserving", using an unstructured mesh, was developed by Pinto [31]. However, this scheme is linked with a precise numbering of the mesh cells that is not compatible with our mesh produced by GMSH. For this reason we have decided to develop a charge-conserving algorithm following Umeda's article but we have also modified it to adapt it to F3MPIC.
The basic idea is to combine two meshes: a cubic one, in which the current deposition is made, and a structured one, in which the field integration phase and the subsequent advancement of the particles are managed.

[1] Generally a Cloud-In-Cell algorithm is a first order assignment consisting in a linear weighting of interesting quantities in the nearest four grid points. The idea consists in representing each macro-particle by a square cloud of size $x \times y$, centered at the particle location and such that its total charge q is uniformly distributed. Then the charge and current densities at each grid point are computed from velocities and positions of the charged particles

3.2 Charge conserving scheme in Particle-In-Cell methods

In this section we show how the zigzag scheme proposed by Umeda is charge conserving. To simplify as much as possible the notation, we analyze only the 1D and 2D case, in fact the 3D case is analog to the 2D case with longer equations.
Achieve charge conserving means to find a method to ensure that the computed charge and the current densities satisfy the discrete version of the continuity equation 3.4. In a 2D cartesian mesh the charge and current densities will be defined, in the following, for all i, j mesh indices, as: $\rho^n_{i,j}, J_x{}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}, J_y{}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$.
We have decided to follow the scheme proposed by Umeda. This scheme can be considered as a generalization of the Villasenor-Buneman's method in which a particle trajectory over one time step is assumed to be a straight line. In Umeda's paper, when a particle moves across the cell meshes, the particle movement is assigned to separate motions in each cell by the cell meshes. From superposition of charge flux, charge conservation can be realized for any particle trajectories made up of straight line segments between any start and end points. A more detailed description of the following demonstration can be found in [10].
Let's start assuming, as in standard PIC literature, that the initial distribution function $f_0(\mathbf{x}, \mathbf{u})$, that fulfills the Vlasov's equation 3.1, could be represented by a superimposition of N macro-particles with positions $\mathbf{x}^\mathbf{0}_\alpha$, momenta $\mathbf{u}^\mathbf{0}_\alpha$ and weights p_α with α the particle index.
Let's call $f^h_0(\mathbf{x}, \mathbf{u})$ the approximation of $f_0(\mathbf{x}, \mathbf{u})$ such that

$$f_0(\mathbf{x}, \mathbf{u}) \sim f^h_0(\mathbf{x}, \mathbf{u}) = \sum_{\alpha=1}^{N} p_\alpha \delta(\mathbf{x} - \mathbf{x}^\mathbf{0}_\alpha)\delta(\mathbf{u} - \mathbf{u}^\mathbf{0}_\alpha) \tag{3.5}$$

where $\delta(*)$ is a Dirac's delta function. This aspect can be easily generalized at any $t > 0$. In fact the particle approximation f^h of f, solution of Vlasov's equation with initial data f^h_0 , can be written for all $t > 0$ as

$$f^h(\mathbf{x}, \mathbf{u}, t) = \sum_{\alpha=1}^{N} p_\alpha \delta(\mathbf{x} - \mathbf{x}_\alpha(t))\delta(\mathbf{u} - \mathbf{u}_\alpha(t)) \tag{3.6}$$

Using last equation, the particle approximation of the charge and the current densities can be written in the following way

$$\begin{aligned}\rho^h(\mathbf{x},t) &= q\sum_{\alpha=1}^{N} p_\alpha S^0(\mathbf{x}-\mathbf{x}_\alpha(t)) = q\sum_{\alpha=1}^{N} p_\alpha \delta(\mathbf{x}-\mathbf{x}_\alpha(t)) \\ \mathbf{J}^h(\mathbf{x},t) &= q\sum_{\alpha=1}^{N} p_\alpha \mathbf{v}_\alpha(t) S^0(\mathbf{x}-\mathbf{x}_\alpha(t)) = q\sum_{\alpha=1}^{N} p_\alpha \mathbf{v}_\alpha(t)\delta(\mathbf{x}-\mathbf{x}_\alpha(t))\end{aligned} \tag{3.7}$$

This weighting is done with the so called "zero-order form or shape factor" $S^0 \equiv \delta$, and it is usually used in 1D problem.

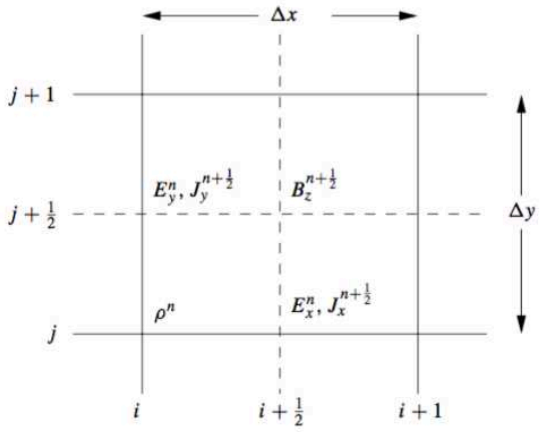


Fig. 3.2: *Locations of the field components on the spatial grid. Figure taken from [10]*

The original PIC methods developed in the 50's were based on using a Dirac's delta also as the shape function in space. But now for the spatial shape functions, all commonly used PIC methods are based on the use of the so-called b-splines. The b-spline functions are a series of consecutively higher order functions obtained from each other by integration. In what follows we will introduce the S^0 and S^1 form factors that we have used in 2D.

Let's now assume that we are looking for an approximation of the Vlasov-Maxwell's system on a bounded domain $\Omega \in \mathbb{R}^2$, in particular consider a grid cell of size $\Delta x \times \Delta y$. In a two dimensional domain, the Maxwell's system consists of two sets of decoupled equations. The first set, usually referred to as the Transverse Electric (TE) mode, involves the (E_x, E_y, B_z) components, whereas the second set, the Transverse Magnetic mode (TM) involves the remaining components, namely (E_z, B_x, B_y). Here we shall only consider the former, since the latter can be dealt with in a similar manner.

The components of electromagnetic fields $(\mathbf{E}, \mathbf{B})$, the charge density ρ and the current densities components J_x, J_y, are located at grid positions as shown in figure 3.2. In particular it is clear as, for example, the current densities components J_x, J_y need to be computed at the fractional grid points $(X_{i+\frac{1}{2}}, Y_j)$ and $(X_i, Y_{j+\frac{1}{2}})$.

This follows from the discrete version of Maxwell's equations 3.2a) and 3.2b) in TE mode, that is given by the standard, second order in time and space, finite difference Yee's scheme

$$\begin{aligned}
\frac{(E_x)^{n+1}_{i+\frac{1}{2},j} - (E_x)^{n}_{i+\frac{1}{2},j}}{\Delta t} &= -\frac{1}{\epsilon_0}(J_x)^{n+\frac{1}{2}}_{i+\frac{1}{2},j} + c^2\frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} - (B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \\
\frac{(E_y)^{n+1}_{i,j+\frac{1}{2}} - (E_y)^{n}_{i,j+\frac{1}{2}}}{\Delta t} &= -\frac{1}{\epsilon_0}(J_y)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - c^2\frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} - (B_z)^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x}
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
\frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} - (B_z)^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta t} = &-\frac{(E_y)^{n}_{i+1,j+\frac{1}{2}} - (E_y)^{n}_{i,j+\frac{1}{2}}}{\Delta x} \\
&+ \frac{(E_x)^{n}_{i+\frac{1}{2},j+1} - (E_x)^{n}_{i+\frac{1}{2},j}}{\Delta y}
\end{aligned} \tag{3.9}$$

Generally each current density and charge density results from the superposition of the charge and density flux contributed by each particle. In a 2D system the simple "zero-order form factor" presented above can be extended to a "first order form factor" called S^1. This first order form factor can be used to generalize equations 3.7. In particular the charge densities and current densities on a 2D grid can be computed as follows

$$\begin{aligned}
(\rho)^{n}_{i,j} &= \sum_{\alpha=1}^{N} qp_\alpha S^1(X_i - x^n_\alpha, Y_j - y^n_\alpha) \\
(J_x)^{n+\frac{1}{2}}_{i,j} &= \sum_{\alpha=1}^{N} qp_\alpha (v_x)^{n+\frac{1}{2}}_\alpha S^1(X_i - x^{n+\frac{1}{2}}_\alpha, Y_j - y^{n+\frac{1}{2}}_\alpha) \\
(J_y)^{n+\frac{1}{2}}_{i,j} &= \sum_{\alpha=1}^{N} qp_\alpha (v_y)^{n+\frac{1}{2}}_\alpha S^1(X_i - x^{n+\frac{1}{2}}_\alpha, Y_j - y^{n+\frac{1}{2}}_\alpha)
\end{aligned} \tag{3.10}$$

In 2D the first order "form factor" S^1 is the tensor product of one dimensional splines such that we can rewrite it as

$$S^1(x, y) = S^1_{\Delta x}(x) S^1_{\Delta y}(y) \tag{3.11}$$

where

$$S^1_k(x) = \begin{cases} \frac{1}{k}(1 - |\frac{x}{k}|) & \text{if} \quad |x| < k \\ 0 \quad \text{else} \end{cases} \tag{3.12}$$

and $k = \Delta x, \Delta y$. A theorem (see [10]) ensures that

$$S^1_k(x) = S^0_k(x) * S^0_k(x) \tag{3.13}$$

and, more generally, if we consider a form factor $S_k^m(x)$ of arbitrary order $m \in \mathbb{N}^*$, holds that

$$S_k^m(x) = S_k^0(x) * S_k^{m-1}(x) = \frac{1}{k} \int_{x-\frac{k}{2}}^{x+\frac{k}{2}} S_k^{m-1}(u)du \tag{3.14}$$

with

$$S_k^0(x) = \begin{cases} \frac{1}{k} & \text{if} \quad |x| < \frac{k}{2} \\ 0 & \text{else} \end{cases} \tag{3.15}$$

Using these equations, we can rewrite $(\rho)_{i,j}^n$ in 3.10 as

$$(\rho)_{i,j}^n = \sum_{\alpha=1}^{N} qp_\alpha S^1(X_i - x_\alpha^n, Y_j - y_\alpha^n) = \sum_{\alpha=1}^{N} qp_\alpha S_{\Delta x}^1(X_i - x_\alpha^n) S_{\Delta y}^1(Y_j - y_\alpha^n) \tag{3.16}$$

Using now equation 3.14 we can rewrite the last one as

$$(\rho)_{i,j}^n = q \sum_{\alpha=1}^{N} p_\alpha \frac{1}{\Delta x \Delta y} \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \int_{Y_{j-\frac{1}{2}}}^{Y_{j+\frac{1}{2}}} S_{\Delta x}^0(x - x_\alpha^n) S_{\Delta y}^0(y - y_\alpha^n) dxdy \tag{3.17}$$

The previous equation shows as the charge density, at a grid point, is given by the sum of all particles contribution.
Let's start now to analyze the current density and let's consider the current density created only by a single particle α. In particular consider an α particle that moves, but remains inside the same mesh boundary.
Following the same logical passages just explained for charge density and with some calculation (see [10]), it is possible to compute the current density $J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}}$ in 3.10 produced by a single α particle that lies inside the same mesh boundary as

$$J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} = \frac{qp_\alpha (v_x^{n+\frac{1}{2}})_\alpha}{\Delta t} \int_{t^n}^{t^{n+1}} S_{\Delta x}^0(X_{i+\frac{1}{2}} - x_\alpha(t)) S_{\Delta y}^1(Y_j - y_\alpha(t)) dt \tag{3.18}$$

where has been introduced a time integral, considering a particle movement between t^n and t^{n+1}. To implement the computation of $J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}}$ in a computer algorithm, it is necessary to solve previous integral.
In this case it is possible to solve it with same passages and to obtain the following result

$$J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} = \frac{qp_\alpha}{\Delta x \Delta y} \frac{x_\alpha^{n+1} - x_\alpha^n}{\Delta t} (1 + j - \frac{y_\alpha^{n+1} + y_\alpha^n}{2\Delta y}) \tag{3.19}$$

Similar equations hold for $J_{x_{i+\frac{1}{2},j+1}}^{n+\frac{1}{2}}, J_{y_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}}, J_{y_{i+1,j+\frac{1}{2}}}^{n+\frac{1}{2}}$.
Until now we have considered only the motion of a particle that always lies inside the same mesh boundary.
In line of principle it should be necessary to analyze all kinds of particles motion; for example it should be necessary to analyze also the current density of a particle

that crosses one face of the mesh and so on. However scope of this section is only to show the charge conservation of this approach, i.e. to show how, for example, 3.18 fulfills the discrete version of the continuity equation. For this reason we will proof only the following proposition referring the reader to [10] to have more details.

Definition 1. *For all $m \in \mathbb{N}^*$ the charge and current densities defined for all i, j by*

$$
\begin{aligned}
(\rho)_{i,j}^{n} &= q \sum_{\alpha=1}^{N} p_\alpha S_{\Delta x}^{m}(X_i - x_\alpha^n) S_{\Delta y}^{m}(Y_j - y_\alpha^n) \\
J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} &= q \sum_{\alpha=1}^{N} p_\alpha \frac{(v_x^{n+\frac{1}{2}})_\alpha}{\Delta t} \int_{t^n}^{t^{n+1}} S_{\Delta x}^{m-1}(X_{i+\frac{1}{2}} - x_\alpha(t)) S_{\Delta y}^{m}(Y_j - y_\alpha(t)) dt \\
J_{y_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}} &= q \sum_{\alpha=1}^{N} p_\alpha \frac{(v_y^{n+\frac{1}{2}})_\alpha}{\Delta t} \int_{t^n}^{t^{n+1}} S_{\Delta x}^{m}(X_i - x_\alpha(t)) S_{\Delta y}^{m-1}(Y_{j+\frac{1}{2}} - y_\alpha(t)) dt
\end{aligned}
\tag{3.20}
$$

satisfy the discrete equation of charge conservation.

Proof.

$$
\begin{aligned}
&\rho_{i,j}^{n+1} - \rho_{i,j}^{n} \\
&= \int_{t^n}^{t^{n+1}} \frac{d}{dt} \rho_{i,j}(t) dt \\
&= \frac{q p_\alpha}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \frac{d}{dt} \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \int_{Y_{j-\frac{1}{2}}}^{Y_{j+\frac{1}{2}}} S_{\Delta x}^{m-1}(x - x_\alpha(t)) S_{\Delta y}^{m-1}(y - y_\alpha(t)) dx dy dt \\
&= \frac{q p_\alpha}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \int_{Y_{j-\frac{1}{2}}}^{Y_{j+\frac{1}{2}}} \frac{d}{dt} (S_{\Delta x}^{m-1}(x - x_\alpha(t)) S_{\Delta y}^{m-1}(y - y_\alpha(t))) dx dy dt \\
&= \frac{q p_\alpha}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} \int_{Y_{j-\frac{1}{2}}}^{Y_{j+\frac{1}{2}}} (-(v_x^{n+\frac{1}{2}})_\alpha \frac{d S_{\Delta x}^{m-1}}{dx}(x - x_\alpha(t)) S_{\Delta y}^{m-1}(y - y_\alpha(t)) \\
&- (v_y^{n+\frac{1}{2}})_\alpha S_{\Delta x}^{m-1}(x - x_\alpha(t)) \frac{d S_{\Delta y}^{m-1}}{dy}(y - y_\alpha(t))) dx dy dt
\end{aligned}
\tag{3.21}
$$

Where the spatial derivative follows from these two equation for $t \in [t^n, t^{n+1}]$

$$
\begin{aligned}
x_\alpha(t) &= x_\alpha^n + (t - t^n)(v_x^{n+\frac{1}{2}})_\alpha \\
y_\alpha(t) &= y_\alpha^n + (t - t^n)(v_y^{n+\frac{1}{2}})_\alpha
\end{aligned}
\tag{3.22}
$$

Following with calculations of 3.21, we obtain

$$\begin{aligned}
&\rho_{i,j}^{n+1} - \rho_{i,j}^{n} = \\
&- \frac{qp_\alpha(v_x^{n+\frac{1}{2}})_\alpha}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{Y_{j-\frac{1}{2}}}^{Y_{j+\frac{1}{2}}} S_{\Delta y}^{m-1}(y - y_\alpha(t)) \\
&\times (S_{\Delta x}^{m-1}(X_{i+\frac{1}{2}} - x_\alpha(t)) - S_{\Delta x}^{m-1}(X_{i-\frac{1}{2}} - x_\alpha(t)))dydt \\
&- \frac{qp_\alpha(v_y^{n+\frac{1}{2}})_\alpha}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{X_{i-\frac{1}{2}}}^{X_{i+\frac{1}{2}}} S_{\Delta x}^{m-1}(x - x_\alpha(t)) \\
&\times (S_{\Delta y}^{m-1}(Y_{j+\frac{1}{2}} - y_\alpha(t)) - S_{\Delta y}^{m-1}(Y_{j-\frac{1}{2}} - y(t)))dxdt
\end{aligned} \tag{3.23}$$

Finally using 3.14 and remembering that $\Delta y \equiv Y_{j+\frac{1}{2}} - Y_{j-\frac{1}{2}}$ and $\Delta x \equiv X_{i+\frac{1}{2}} - X_{i-\frac{1}{2}}$ we obtain

$$\begin{aligned}
&\rho_{i,j}^{n+1} - \rho_{i,j}^{n} = \\
&- \frac{qp_\alpha(v_x^{n+\frac{1}{2}})_\alpha}{\Delta x} \int_{t^n}^{t^{n+1}} S_{\Delta y}^{m}(Y_j - y_\alpha(t)) \\
&\times (S_{\Delta x}^{m-1}(X_{i+\frac{1}{2}} - x_\alpha(t)) - S_{\Delta x}^{m-1}(X_{i-\frac{1}{2}} - x_\alpha(t)))dt \\
&- \frac{qp_\alpha(v_y^{n+\frac{1}{2}})_\alpha}{\Delta y} \int_{t^n}^{t^{n+1}} S_{\Delta x}^{m}(X_i - x_\alpha(t)) \\
&\times (S_{\Delta y}^{m-1}(Y_{j+\frac{1}{2}} - y_\alpha(t)) - S_{\Delta y}^{m-1}(Y_{j-\frac{1}{2}} - y_\alpha(t)))dt
\end{aligned} \tag{3.24}$$

Using definitions in 3.20 we can then write the last equation as

$$\begin{aligned}
&\rho_{i,j}^{n+1} - \rho_{i,j}^{n} \\
&= -\Delta t \left(\frac{1}{\Delta x}(J_{x}{}^{n}_{i+\frac{1}{2},j} - J_{x}{}^{n}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(J_{y}{}^{n}_{i,j+\frac{1}{2}} - J_{y}{}^{n}_{i,j-\frac{1}{2}}) \right)
\end{aligned} \tag{3.25}$$

that is the discrete version of the continuity equation

□

3.3 Zigzag scheme for charge conserving

Having obtained a formulation that ensures the conservation of charge we now show the practical algorithm proposed by Umeda to compute integrals in 3.20.

In three dimensions the difference form of the continuity equation is a simple extension of 3.25:

$$\frac{J_x^{t+\frac{\Delta t}{2}}(i+\frac{1}{2},j,k)-J_x^{t+\frac{\Delta t}{2}}(i-\frac{1}{2},j,k)}{\Delta x}+\frac{J_y^{t+\frac{\Delta t}{2}}(i,j+\frac{1}{2},k)-J_y^{t+\frac{\Delta t}{2}}(i,j-\frac{1}{2},k)}{\Delta y}$$
$$+\quad\frac{J_z^{t+\frac{\Delta t}{2}}(i,j,k+\frac{1}{2})-J_z^{t+\frac{\Delta t}{2}}(i,j,k-\frac{1}{2})}{\Delta z}=\frac{\rho^t(i,j,k)-\rho^{t+\Delta t}(i,j,k)}{\Delta t} \tag{3.26}$$

In the last equation, if compared with 3.25, we have introduced a new index k to identify mesh cells and, for clarity, we have indicated the time discretization with t and $t+\frac{\Delta t}{2}$ rather than n and $n+\frac{1}{2}$, avoiding also to use index notation for grid indices i.e the previous $J_{i,j,k}$ is now $J(i,j,k)$.
Let's now consider a particle of charge q moving from (x^t,y^t,z^t) to $(x^{t+\Delta t},y^{t+\Delta t},z^{t+\Delta t})$, it is possible to introduce the following notation to represent particle movement

$$\begin{aligned} x_1&=x^t & y_1&=y^t & z_1&=z^t \\ x_2&=x^{t+\Delta t} & y_2&=y^{t+\Delta t} & z_2&=z^{t+\Delta t} \end{aligned} \tag{3.27}$$

If we introduce a structured cubic mesh composed of cubes of side lenght $\Delta x,\Delta y$ and Δz, it is possible to track initial and final particle position (i.e. to find index of cubic mesh in which the particle is located) using standard scientific C function FLOOR in the following way

$$\begin{aligned} i_1&=\text{FLOOR}(\frac{x_1}{\Delta x}) & j_1&=\text{FLOOR}(\frac{y_1}{\Delta y}) & k_1&=\text{FLOOR}(\frac{z_1}{\Delta z}) \\ i_2&=\text{FLOOR}(\frac{x_2}{\Delta x}) & j_2&=\text{FLOOR}(\frac{y_2}{\Delta y}) & k_2&=\text{FLOOR}(\frac{z_2}{\Delta z}) \end{aligned} \tag{3.28}$$

Especially i_l,j_l and k_l with $l=1,2$ are the largest integer value not greater than $\frac{x_l}{\Delta x}$, $\frac{y_l}{\Delta y}$ and $\frac{z_l}{\Delta z}$. To preserve mesh compatibilities, as in the previous MCC code, it is necessary that particle does not move more than grid spacing $\Delta x,\Delta y$ and Δz for one time step Δt i.e.

$$\begin{aligned} v_x\Delta t&<\Delta x \\ v_y\Delta t&<\Delta y \\ v_z\Delta t&<\Delta z \end{aligned} \tag{3.29}$$

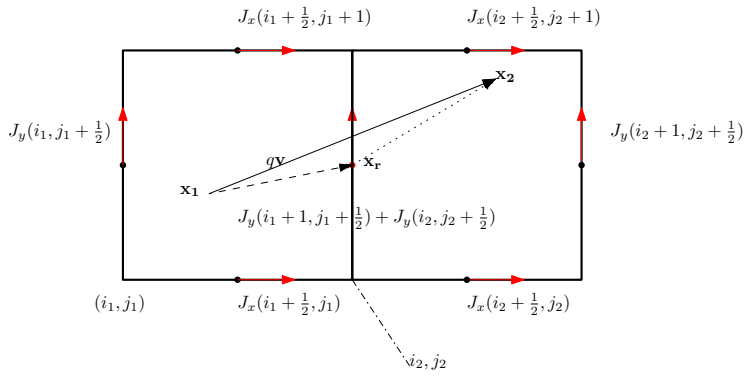


Fig. 3.3: *Current decomposition in a 2D mesh. In this case we have analyzed the case in which the particle crosses a mesh boundary. It is interesting to note that the vectors, used for the decomposition, follow the direction of the particle motion. The 3D case follows in a straight-forward manner*

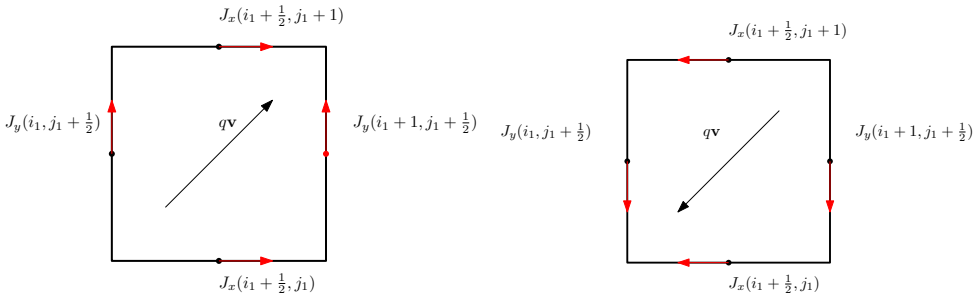
When a particle remains in the same cell of the square mesh during its movement, a charge flux of the particle can be computed, from the start point (x_1, y_1, z_1) to the end point (x_2, y_2, z_2) of the particle movement, with the procedure described in [29] and introduced in the previous section. However particle can cross mesh boundaries, for this reason we follow the procedure proposed by Umeda in [37],[38] to compute charge flux, adapting it to our unstructured mesh. The basic idea to obtain a charge conserving method is to decompose the particle movement when the particle crosses mesh boundaries. The decomposition is done with a special assignment pattern; in this one the flux $q\mathbf{v} = q(v_x, v_y, v_z)$ of the particle that crosses the boundary is decomposed into two segments: $\mathbf{F_1} = (F_{x1}, F_{y1}, F_{z1})$ and $\mathbf{F_2} = (F_{x2}, F_{y2}, F_{z2})$. To obtain a consistent decomposition it is necessary to introduce a third point called "relay point" with coordinates (x_r, y_r, z_r), as shown in figure 3.3. $\mathbf{F_1}$ is the particle flux between (x_1, y_1, z_1) and (x_r, y_r, z_r), while $\mathbf{F_2}$ between (x_r, y_r, z_r) and (x_2, y_2, z_2).
In this way it is possible to define $\mathbf{F_1}$ and $\mathbf{F_2}$ (and their components) in the following way

$$\begin{aligned} F_{x1} &= q\frac{x_r - x_1}{\Delta t} & F_{y1} &= q\frac{y_r - y_1}{\Delta t} & F_{z1} &= q\frac{z_r - z_1}{\Delta t} \\ F_{x2} &= q\frac{x_2 - x_r}{\Delta t} & F_{y2} &= q\frac{y_2 - y_r}{\Delta t} & F_{z2} &= q\frac{z_2 - z_r}{\Delta t} \end{aligned} \tag{3.30}$$

We assign these computed charge flux values to the adjacent grid points (in a full $3D$ configuration to 24 grid points, 12 if we consider a particle that remains in the same cube) using the following first-order shape factor defined at the midpoints $\frac{(x_1,y_1,z_1)+(x_r,y_r,z_r)}{2}$ and $\frac{(x_2,y_2,z_2)+(x_r,y_r,z_r)}{2}$ as

$$\begin{aligned} W_{x1} &= \frac{x_r + x_1}{2\Delta x} - i_1 & W_{y1} &= \frac{y_r + y_1}{2\Delta y} - j_1 & W_{z1} &= \frac{z_r + z_1}{2\Delta z} - k_1 \\ W_{x2} &= \frac{x_2 + x_r}{2\Delta x} - i_2 & W_{y2} &= \frac{y_2 + y_r}{2\Delta y} - j_2 & W_{z2} &= \frac{z_2 + z_r}{2\Delta z} - k_2 \end{aligned} \tag{3.31}$$

If the charge flux is within a single cell mesh the decomposition is always done, but in this case the relay point is inside the cell mesh. This fact is necessary to obtain a recursive algorithm. See for example figure 3.4.



(a)Example of current decomposition in 2D mesh

(b)Example of current decomposition in 2D mesh

Fig. 3.4: *Left and Right: Current decomposition in a 2D mesh. In this case we have analyzed the case in which the particle remains in the same mesh boundaries during its motion. It is interesting to note that the vectors, used for the decomposition, follow the direction of the particle motion. The 3D case follows in a straight-forward manner*

In figure 3.3 we have plotted the decomposition of the charge flux for a particle that crosses a mesh boundary; in this case we have used two vectors. More generally a charge flux can be decomposed into n different segments depending on the location of start and end points. In a cubic mesh n varies from 1 to 4. To obtain a model that is as complete as possible, it should be necessary to analyze all these cases, however in Umeda's formulation a charge flux is always decomposed in only two flux vectors regardless of the start and end points, and a relay point is always introduced.
In particular the choice of the relay point in a consistent manner is crucial to obtain a recursive and adaptable algorithm. In [37],[38] there is the following proposal for the relay point (x_r, y_r, z_r)

$$\mathbf{x}_r = \begin{cases} x_r = \min\{\min(i_1\Delta x, i_2\Delta x) + \Delta x, \max[\max(i_1\Delta x, i_2\Delta x), \frac{x_1+x_2}{2}]\} \\ y_r = \min\{\min(j_1\Delta y, j_2\Delta y) + \Delta y, \max[\max(j_1\Delta y, j_2\Delta y), \frac{y_1+y_2}{2}]\} \\ z_r = \min\{\min(k_1\Delta z, k_2\Delta z) + \Delta z, \max[\max(k_1\Delta z, k_2\Delta z), \frac{z_1+z_2}{2}]\} \end{cases} \tag{3.32}$$

Using equations 3.31, 3.30 and the relay point just defined, it is possible to assign the charge fluxes $\mathbf{F_1}$ and $\mathbf{F_2}$ in a $3D$ configuration to 24 grid points.
Once the charge fluxes are computed, Umeda proposed the following scheme to compute the current density vector

$$\begin{aligned}
J_x(i_l+\frac{1}{2}, j_l, k_l) &= \frac{1}{\Delta x\Delta y\Delta z}F_{xl}(1-W_{yl})(1-W_{zl}) \\
J_x(i_l+\frac{1}{2}, j_l+1, k_l) &= \frac{1}{\Delta x\Delta y\Delta z}F_{xl}W_{yl}(1-W_{zl}) \\
J_x(i_l+\frac{1}{2}, j_l, k_l+1) &= \frac{1}{\Delta x\Delta y\Delta z}F_{xl}(1-W_{yl})W_{zl} \\
J_x(i_l+\frac{1}{2}, j_l+1, k_l+1) &= \frac{1}{\Delta x\Delta y\Delta z}F_{xl}W_{yl}W_{zl} \\
J_y(i_l, j_l+\frac{1}{2}, k_l) &= \frac{1}{\Delta x\Delta y\Delta z}F_{yl}(1-W_{xl})(1-W_{zl}) \\
J_y(i_l+1, j_l+\frac{1}{2}, k_l) &= \frac{1}{\Delta x\Delta y\Delta z}F_{yl}W_{xl}(1-W_{zl}) \\
J_y(i_l, j_l+\frac{1}{2}, k_l+1) &= \frac{1}{\Delta x\Delta y\Delta z}F_{yl}(1-W_{xl})W_{zl} \\
J_y(i_l+1, j_l+\frac{1}{2}, k_l+1) &= \frac{1}{\Delta x\Delta y\Delta z}F_{yl}W_{xl}W_{zl} \\
J_z(i_l, j_l, k_l+\frac{1}{2}) &= \frac{1}{\Delta x\Delta y\Delta z}F_{zl}(1-W_{xl})(1-W_{yl}) \\
J_z(i_l+1, j_l, k_l+\frac{1}{2}) &= \frac{1}{\Delta x\Delta y\Delta z}F_{zl}W_{xl}(1-W_{yl}) \\
J_z(i_l, j_l+1, k_l+\frac{1}{2}) &= \frac{1}{\Delta x\Delta y\Delta z}F_{zl}(1-W_{xl})W_{yl} \\
J_z(i_l+1, j_l+1, k_l+\frac{1}{2}) &= \frac{1}{\Delta x\Delta y\Delta z}F_{zl}W_{xl}W_{yl}
\end{aligned} \tag{3.33}$$

with $l = 1, 2$.
A little inspection of these equations shows that these ones are the same equations proposed in the previous section to compute the current density analyzing the case in which the particle is always within the same cell of the mesh; there is only a redefinition of the various involved peaces.
For example a simple comparison between equation 3.19 that we report here for clarity: $J_{x_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} = \frac{qp_\alpha}{\Delta t \Delta y} \frac{x_\alpha^{n+1} - x_\alpha^n}{\Delta x}(1 + j - \frac{y_\alpha^{n+1} + y_\alpha^n}{2\Delta y})$ and equation 3.33a) shows that, leaving out the third term $(1 - W_{zl})$ in 3.33a) due to the 3D extension, and looking at the definitions of W_{yl} and F_{xl} in 3.30 and 3.31, equations 3.19 and 3.33a) are exactly the same, with the adaptation of the relay point to this case. This ensures as the algorithm proposed by Umeda is well posed to achive charge conservation.

3.3.1 Integration of the zigzag scheme in F3MPIC

In the previous section we have shown how to compute the contribution to the current density vector using a single charged particle. The algorithm must be repeated for all the used particles in the system at each time step.
However the main disadvantage of the algorithm just proposed is connected with the fact that the new algorithm is only valid in a structured cubic mesh.
As stated, F3MPIC uses an unstructured cubic mesh and for this reason it is necessary to adapt the previous formulation to our case. If we consider a domain of characteristic length L_x, L_y and L_z, along each of these directions, the number of cubes of the cubic mesh can be computed with the following equation

$$N_i = \frac{L_i}{\Delta i} \tag{3.34}$$

with $i = x, y$ or z. From this, the number of nodes along each direction is $(2N_i + 1)$. In this way the number of total nodes necessary to compute are

$$N_{\text{Total nodes}} = (2N_x + 1)(2N_y + 1)(2N_z + 1) \tag{3.35}$$

The first disadvantage to solve is to find a method to deposit the current in half indices nodes (like $\frac{1}{2}$). To solve this, we have decided to construct a new cubic mesh called in the following "FALSE mesh" starting from the previous one that will be called "TRUE mesh".
The nodes of the new mesh are defined, thanks to a translation, in the following way:

$$\mathbf{x}_{\text{FALSE}} = 2\mathbf{x}_{\text{TRUE}} + \mathbf{tr} \tag{3.36}$$

where $\mathbf{tr}$ is a translation vector defined as

$$\mathbf{tr} \equiv (2N_x, 2N_y, 2N_z) \tag{3.37}$$

and with $\mathbf{x}_{\text{FALSE/TRUE}}$ the vector of the three indices of the cubic mesh (i_s, j_s, k_s) respectively for the mesh state s=FALSE and s=TRUE.
In this way all the indices for the FALSE mesh are positive integers and a storing in a pointer like quantity can be possible. In particular, given particle positions i.e. given the start and the end positions of a particle, these positions are always used to find, in the cubic "TRUE"

mesh, the cube in which the particle is located and in which it is necessary to compute and to deposit the current density vector, in accordance with 3.28 and 3.33.
Once the current is computed, using equation 3.36, the "FALSE" mesh cube associated to the just computed "TRUE" mesh cube is found, and the current is saved under the integer nodes of this "FALSE" cube.
For example the component J_x saved before in the TRUE half mesh node $(i_{l\text{TRUE}} + \frac{1}{2}, j_{l\text{TRUE}} + 1, k_{l\text{TRUE}})$ (with $l = 1, 2$), is now saved under the FALSE mesh indices $(2i_{l\text{TRUE}} + 1 + tr_x, 2j_{l\text{TRUE}} + 2 + tr_y, 2k_{l\text{TRUE}} + tr_z)$ and so on.
In what follows we will not make more distinctions between the "TRUE" mesh indices and the "FALSE" ones.

3.3.2 Interpolation of current density on F3MPIC nodes

In the previous sections we have shown how to compute the current density vector on nodes of the structured mesh. In particular we have shown how a particle contributes to the vector **J** on twelve points of the structured mesh (i.e. on twelve points of the cube of the structured mesh in which the particle is located).
The next step necessary to validate the just presented algorithm, it is to plot at various time steps the vector **J** in module and direction. **J** is a punctual quantity and for this reason it is important to decide where to plot the current density vector. To obtain a result directly comparable with other quantities computed in F3MPIC, we have decided to compute the current density vectors on F3MPIC nodes. It follows that it is necessary to find a method to interpolate the values of the current density, that are known on the structured mesh points, on the points of the unstructured F3MPIC mesh. Fortunately the tracking operation of F3MPIC nodes within the structured mesh can be done in an easy way: it is sufficient to repeat what was done in previous sections with the FLOOR function using, this time, the node coordinates as the coordinates of "a particle".
Once found the structured mesh cube in which the F3MPIC node is located, it is now important to specify the interpolating method used. In particular in this cube there are twelve points in which the current density vector has been computed (corresponding to points with integer and semi-integer indices). To obtain the value of the current density vector in the F3MPIC node, that is located inside this cube, we have decided to use an Inverse Distance Weighting (IDW) interpolating function.
Inverse Distance Weighting (IDW) (see [11],[4])[2] is a type of deterministic method for multivariate interpolation with a known scattered set of points. The assigned values to unknown point is calculated with a weighted average of the values available at the known points.
A general form of finding an interpolate value u at a given point $\mathbf{x} \equiv (x, y, z)$ based on samples $u_i = u_i(\mathbf{x_i})$ of known points $\mathbf{x_i}$ with $i = 1, 2, ...12$ using IDW, is an interpolating function

$$u(\mathbf{x}) = \begin{cases} \frac{\sum_{i=1}^{12} w_i(\mathbf{x}) u_i}{\sum_{i=1}^{12} w_i(\mathbf{x})}, & \text{if } d(\mathbf{x}, \mathbf{x_i}) \neq 0 \text{ for all } i \\ u_i, & \text{if } d(\mathbf{x}, \mathbf{x_i}) = 0 \text{ for same } i \end{cases} \tag{3.38}$$

where w_i is defined as

$$w_i = \frac{1}{d(\mathbf{x}, \mathbf{x_i})^p} \tag{3.39}$$

[2] See also https://en.wikipedia.org/wiki/Inverse_distance_weighting

In particular $\mathbf{x}$ denotes an interpolated (arbitrary) point, $\mathbf{x_i} \equiv (x_i, y_i, z_i)$ is an interpolating (known) point, $d(\mathbf{x}, \mathbf{x_i})$ is a given distance (metric operator) from the known point $\mathbf{x_i}$ to the unknown point $\mathbf{x}$

$$d(\mathbf{x}, \mathbf{x_i}) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{3.40}$$

and 12 is the total number of known points used in interpolation. p is a positive number (> 0). While any p value convenient for a given application may be used, common practice is to use distance ($p = 1$) or distance squared ($p = 2$).
This operation was repeated for all F3MPIC nodes, in such a way at the end of the procedure, the current density vector is known in all these nodes and can be easily studied.
The results obtained and the goodness of the model implemented are presented in the next section. We conclude this section noting the analogy between equation 3.38 and the weighting media equation, in fact the usual weighting media formula for a set of data x_i with $i = 1....N$ with error σ_i is given by

$$\bar{x} = \frac{\sum_{i=1}^{N} \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^{N} \frac{1}{\sigma_i^2}} \tag{3.41}$$

The total error is then given by

$$\sigma_{\bar{x}} = \sqrt{\frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma_i^2}}} \tag{3.42}$$

In equation 3.38 however $d(\mathbf{x}, \mathbf{x_i})$ is not the error associated to the quantity u_i from a formal point of view,[3] however if we want to keep the analogy with standard weighting media equation, it is possible to think $d(\mathbf{x}, \mathbf{x_i})$ as a measure of uncertainty associated by the interpolation to the quantity u_i.
Treating this like an usual error, if we compare equation 3.41 with 3.38, it is clear as $w_i = \frac{1}{d(\mathbf{x}, \mathbf{x_i})} \equiv \frac{1}{\sigma_i^2}$ and $x_i \equiv u_i$ and for this reason $\sigma_i^2 \equiv d(\mathbf{x}, \mathbf{x_i})$. From this comparison it is clear as the total error on the interpolated quantity decreases drastically reducing the dimensions of the sides of the cubes which constitute the mesh; in this way the distance $d(\mathbf{x}, \mathbf{x_i})$ decreases and thus the total error on the interpolated quantity is smaller. This fact has a crucial importance, in fact a large mesh reduces considerably the number of calculations but, at the same time, the interpolated quantities are not fully satisfactory having a large error.

[3] See also as a dimensional analysis of the problem is not strictly fulfilled if we consider $d(\mathbf{x}, \mathbf{x_i})$ as the error associated to the quantity u_i

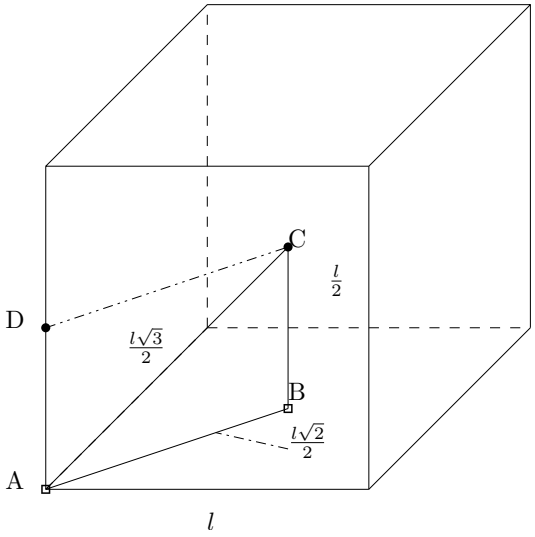


Fig. 3.5: *Cube of structured mesh with edge l*

Starting from these considerations, it is possible to associate an informal error to the quantity $u(\mathbf{x})$ using the following equation

$$\sigma_{u(\mathbf{x})} = \sqrt{\frac{1}{\sum_{i=1}^{12} \frac{1}{d(\mathbf{x},\mathbf{x_i})}}} \tag{3.43}$$

Using a cubic mesh it is also simple to give an estimate of the last quantity. See figure 3.5. If we consider a point in the center of the cube, the error of the interpolation 3.43 is given by

$$\sigma_{u(\mathbf{x})} = \sqrt{\frac{1}{8\frac{1}{\frac{l\sqrt{3}}{2}} + 4\frac{1}{\frac{l\sqrt{2}}{2}}}} = \sqrt{\frac{l\sqrt{6}}{16\sqrt{2}+8\sqrt{3}}} \tag{3.44}$$

As stated the error on the interpolated quantity grows with l. If we consider only the eight cube vertices and we don't consider the points that correspond to semi-integers indices located in the middle of each cube edge, the previous equation can be rewritten as

$$\sigma_{u(\mathbf{x})} = \sqrt{\frac{1}{8\frac{1}{\frac{l\sqrt{3}}{2}}}} = \sqrt{\frac{l\sqrt{3}}{16}} \tag{3.45}$$

As mentioned, this error is not however consistent from a formal point of view, this discussion only wants to show how the size of the mesh is important in the standard PIC formulation. The real, correct error associated to quantity 3.38 must be computed using the standard error propagation formula.

3.4 Computational results and validation of the new algorithm

In this section we show the results obtained for the calculation of the new current density vector. We have decided to study in detail the characteristic properties of the magnetized plasma column in the APEL-device (Applied Plasma Experiment on Linear device). This device was studied at the Institute for Plasma Research in Bhat, Gandhinagar, Gujarat, India and it was also studied at CISAS, a group of Padua university. In figure 3.6 it is possible to see the experimental setup. See [23] to have more details.

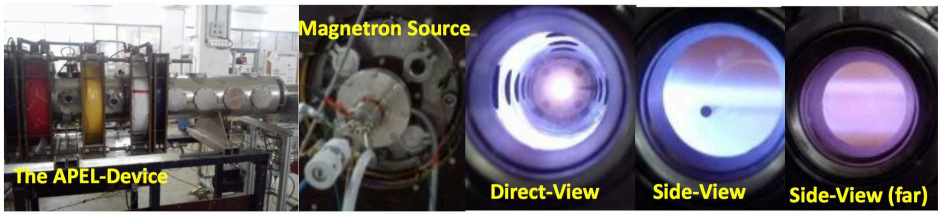


Fig. 3.6: *Image of APEL device (Applied Plasma Experiment on Linear device)*

The plasma source is based on D.C Magnetron discharge produced between a hollow cylindrical cathode of diameter $5.0cm$, length $10cm$ and a differentially pumped constricted hollow anode placed at one end of the cathode. The plasma thus created inside the cathode expands into a plasma column that extends up to a distance of $50cm$ from the source due to the presence of uniform axial magnetic field over that range. In figure 3.7 we have reproduced a section of the APEL experimental setup of figure 3.6. Measurement of radial plasma parameters, using planar Langmuir probe, shows an off-centered density and plasma potential peaking at radial distance of $20 - 25mm$ off-axis from the center. The peak electron density is observed on the order of $10^{16/17}m^{-3}$ at operating power levels of $600W$. The plasma potential follows the characteristic density profile in good agreement with the theoretical model based on radial ambipolar diffusion in the cylindrical column. Measurements of electron saturation current, using planar directional probes, suggest the presence of counter propagating $\mathbf{E} \times \mathbf{B}$ drifted electrons on either side of the density peak. The drift region is also characterized by strong fluctuations in the plasma density and floating potential oscillations that indicate towards drift instability in the magnetized plasma column.
To deep test the new algorithm to compute the current density vector, we have done some simulations using the new APEL system.
To have sustainable calculation times we have simulated a plasma with a mean density of $10^{12/13}m^{-3}$, lower than the experimental one. In fact the aim of these tests is only to see the goodness of the new algorithms implemented and not to compare the simulated results with the unknown experimental ones. In particular we are mainly interested in the agreement between the simulation results and the theoretical predictions. We also have lowered the potential difference of figure 3.6 at $200V$ in fact, working at lower density, it is anyway necessary to maintain valid the compatibility with Debye's length.

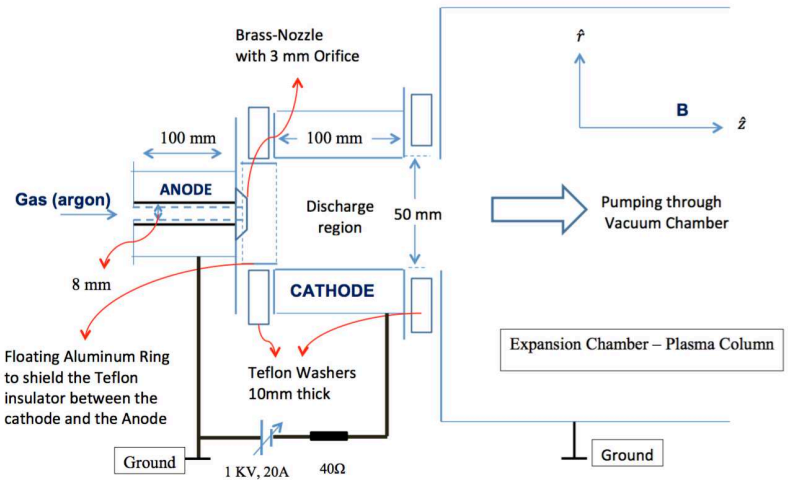


Fig. 3.7: *APEL experimental setup. This setup has been reproduced numerically and it has been studied with a deep simulation campaign. This figure shows the cross-sectional view of the Cylindrical Magnetron Source. In particular it is shown the constricted anode (tube) through which the gas is introduced. Both Anode and Cathode are made of S.S 304 and are water cooled. The pressure inside the cathode is* $0.8/1Pa$*. The maximum axial magnetic field strength is* $30m$ *Tesla.*

Let's start with the compute of the vector **J**.
To better appreciate **J** variation, we only have worked with the imposed electric field (due to a potential difference of 200V), see figure 3.8.
The peak value of the electric field is

$$|\mathbf{E}| \sim 1 * 10^4 \frac{N}{C} \tag{3.46}$$

With a little analysis it is easy to see that this value is fully compatible with the expectations. In fact the system near the two electrodes can be approximated as a plate capacitor with parallel faces of useful area[4]

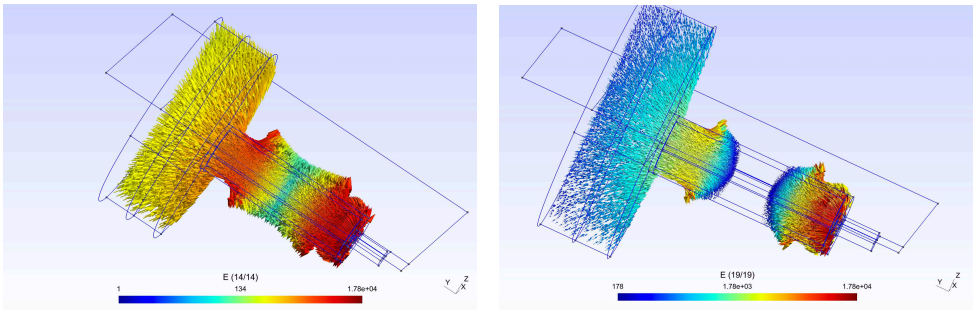
$$\Delta s = \pi r^2 \tag{3.47}$$

and if we call d the distance between the two electrodes, that can be estimated as $d \sim 0,025m$, it is possible to estimate the electric field between the plates as follows

$$E \equiv |\mathbf{E}| = \frac{\Delta V}{d} \sim 8 * 10^3 \frac{N}{C} \tag{3.48}$$

where $\Delta V = 200V$.
This value is fully compatible with the simulated one obtained in figure 3.8. In the same figure it also possible to appreciate as plasma bulk is essentially free from electric field. The scale is always logarithmic to appreciate better the variation of the electric field.

[4] with r the radius of the cylinder of the discharge region i.e. $r = 0,025m$



(a)Electric field configuration.

(b)Electric field configuration with a cut-off on the lowest value that the field can assume

Fig. 3.8: *Electric field configuration in the APEL-device. The electric field is due to a potential different of $200V$, described in figure 3.7. The used scale is logarithmic. In figure b) we have performed a cut-off on the lowest value that the electric field can assume, in this way it is possible to appreciate how plasma bulk is almost free from electric field.*

For what follows it is also interesting to give a rough estimate of electron and ion velocities. Continuing to treat the system as a capacitor, electrons and ions are accelerated of a quantity in module equal to

$$a = \frac{Eq}{m} \tag{3.49}$$

If we consider an initial thermal velocity given by $v_0 = \sqrt{\frac{3k_B T}{m}}$ with an electron temperature of $T_e \sim 11600K$, and an ion temperature of $T_i \sim 300K$, and also if we consider an integration time of $\Delta t \sim 10^{-9}s$, the exit electrons and ions speeds could be estimated as

$$v \sim v_0 + \frac{Eq}{m}\Delta t \tag{3.50}$$

Using the value for E obtained in figure 3.8, it is easy to see that $v_e \sim 10^6 \frac{m}{s}$ and $v_i \sim 10^3 \frac{m}{s}$.
Let's continue now with the validation of the new **J** compute.
To validate the new algorithm it is necessary to compare the simulated direction and module of **J**, with a rough estimate of the same vector done with some simple calculations.
The theoretical trend of **J** can be obtained using the following equation

$$\mathbf{J} \equiv \mathbf{J}_e + \mathbf{J}_{ion} = n_e q_e \mathbf{v_e} + n_i q_i \mathbf{v_i} \tag{3.51}$$

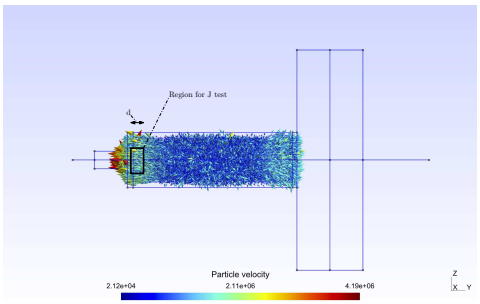
where $q_e = -q_i = -1,6022 * 10^{-19}C$, $\mathbf{v_e}, \mathbf{v_i}$ are the ions and electrons velocities and n_e, n_i are the number of electrons and ions per unit volume ("number density") (SI unit: m^{-3}).
Working at equal electron and ion density, the most important contribution for the calculation of the current density vector is given by the more mobile species i.e. the species with a higher speed (in our case electrons). In such a way if the algorithm implemented works well, the vector **J** should be in the opposite direction of the electric speed.
Equation 3.51 also ensures a way to obtain an estimate of the module of **J**, in fact chosen a fixed point $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ the current density vector module in that point can be computed as

$$J(x,y,z) \equiv |\mathbf{J}(\mathbf{x},\mathbf{y},\mathbf{z})| = |\mathbf{J}_e(\mathbf{x},\mathbf{y},\mathbf{z}) + \mathbf{J}_{ion}(\mathbf{x},\mathbf{y},\mathbf{z})| \sim n_e(\mathbf{x},\mathbf{y},\mathbf{z}) q_e |\mathbf{v_e}(\mathbf{x},\mathbf{y},\mathbf{z})| \tag{3.52}$$

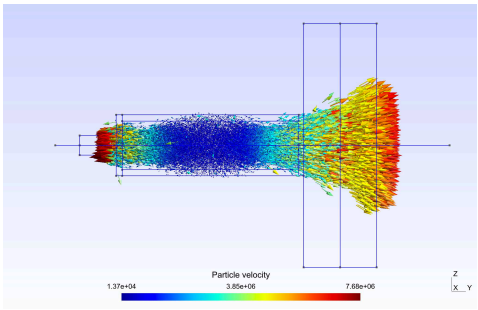
However we work with current density vector computed on mesh nodes, in which we don't know the exact value of electron speed.

For this reason we have chosen a fixed region as the one in figure 3.9a), and for the estimates of n_e and $\mathbf{v_e}$, we have used mean quantities. In such a way the mean current density vector, in a fixed region, can be computed as

$$< J >\equiv< \mathbf{J}(\mathbf{x},\mathbf{y},\mathbf{z}) >_{(\mathbf{x},\mathbf{y},\mathbf{z})\in\text{chosen region}} \tag{3.53}$$



(a)Argon electron velocities immediately after the initial loading



(b)Argon electron velocities after few global time-steps.

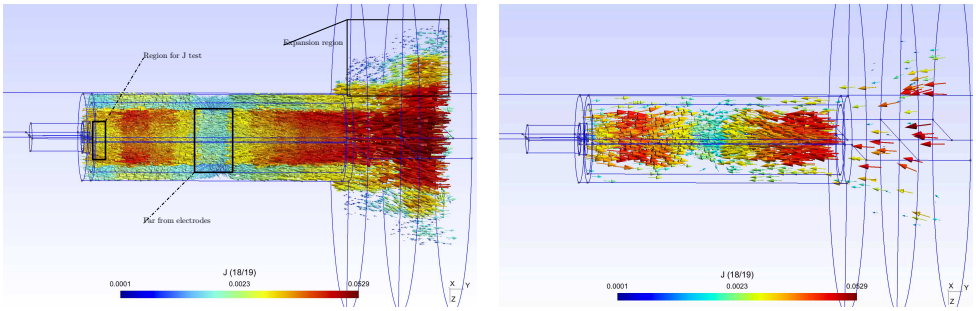
Fig. 3.9: *Argon electrons velocities. In figure a) it is possible to appreciate Argon electrons velocities at the beginning of the simulation (immediately after the initial loading). Figure b) is the same of figure a) but in this case electron velocities are saved after few global time-steps. The electrons are accelerated thanks to the potential difference. In particular it is interesting to note as the maximum electron velocity is fully compatible with the expected one computed in this section. In figure a) we have underlined a region. This region is the region that was chosen to perform a test to validate the compute of the current density module.*

We have decided to use this region because this region is near the electrodes, where, therefore, the plasma shielding effects are less relevant and in which we have a greater control on electron energy and density. In the expansion region there are strong fluctuations in electron density and, for this reason, it isn't simple to obtain here a consistent estimate of $\mathbf{J}$.
The estimation of the current density vector in this region is

$$J = 0.02\frac{\mathrm{A}}{\mathrm{m}^2} \tag{3.54}$$

It is obvious that the result obtained can not be 100% reliable, in fact this value is a mean value while $\mathbf{J}$ is a punctual quantity computed in a precise point i.e. $\mathbf{J} \equiv \mathbf{J}(\mathbf{x},\mathbf{y},\mathbf{z})$, however the obtained result gives an idea of the order of magnitude of $\mathbf{J}$.
If we compare the obtained theoretically result with the computational $\mathbf{J}$ vector in figure 3.10, it is clear as there is a good agreement between the predicted result and the computed one both in module and direction.
In the same figure it is also possible to see as the current density vector is smaller in two particular regions: in the middle of the discharge region i.e. in plasma bulk and in the expansion region.
In the former case the vector $\mathbf{J}$ is calculated far from electrodes where the electric field is partially screened and electrons have lower velocity. In fact as shown in figure 3.9, electron velocity in that region is lower of two order of magnitude if compared with the same value computed near the electrodes. It follows that the same module of the current density vector,

being directly proportional to the module of the electron speed, is smaller of two orders of magnitude if compared with the same value computed near the electrodes. In the latter the vector **J** is computed in a region in which plasma electron density is very low. It follows that the module of the current density ranges for different orders of magnitude.
For this reason in figure 3.10a) and b) we have used a logarithmic scale to analyze **J** variation, in particular figure 3.10b) is the same of figure 3.10a), but with a less number of plotted points to appreciate better the direction of **J**.



(a)Current density vector in the APEL device

(b)Same of figure a) but with a reduced number of plotted points

Fig. 3.10: *Current density vector. The simulated* **J** *is in agreement both in module and direction with the expected one. In figure a) we have underlined two regions in which the module of the current density vector is lower. These regions correspond to plasma bulk and the expansion region. In the first region the electron velocity is lower and so the current density module. In the second region there are strong fluctuations of plasma density. This fact is responsible of the small* **J** *values. Figure b) is the same of figure a) but, in this case, we have reduced the number of plotted points to appreciate the direction of* **J**. *Both figures use a logarithmic scale to better appreciate the current density variations*

3.5 Conclusions

In this section we have introduced in F3MPIC a new algorithm to ensure charge conservation. In particular we have introduced a new formulation following Umeda's paper.
After an initial detailed framework to explain in detail why this new formulation preserves the continuity equation, we have developed a new structured mesh and used it to deposit the current density vector. In particular, we have explained how to save quantities in half-integer points using a FALSE structured mesh defined thanks to a translation.
By interpolation we have then computed the current density vector on F3MPIC nodes starting from the current density values saved in structured mesh nodes. The obtained results were then compared with the expected ones using as test geometry an innovative device called APEL device. In particular, simulations show that the computed current density vector is compatible with the expected one both in direction and module, showing the goodness of the implemented algorithm.

4 A new particle tracking method in unstructured grid

> *Each piece, or part, of the whole of nature is always merely an approximation to the complete truth, or the complete truth so far as we know it. In fact, everything we know is only some kind of approximation because we know that we do not know all the laws as yet.*
>
> **— Richard P. Feynman**

Abstract

The numerical simulation of Lagrangian particles by means of PIC techniques typically involves the tracking of the particles within the spatial domain of the problem. In the case of structured cartesian grids, the particle-locating problem can be readily solved. However, this does not hold for the general case of unstructured grids, which are nowadays widely used for complex geometries, such as those encountered in plasma applications. In this chapter we will implement and test a new particles tracking algorithm proposed recently by Haselbacher A. and others (see [18]). In particular we have substituted the old F3MPIC particle tracking with this new one, integrating the new algorithm with a completely revisited charge deposition algorithm to manage boundaries crossing, internal deposition and secondary electron emission.

4.1 Introduction

Particle tracking is of key importance for quantitative analysis of PIC dynamic processes. Because manually detecting and following large numbers of individual particles is not feasible, automated computational methods have been developed for these tasks by many research groups. The problem of localization of particles within an unstructured mesh is highly multidisciplinary, in fact a "particle" may be anything: from a single electron to a macromolecular complex, organelle, virus or microsphere, and the task of detecting and following individual particles in a time series of images (for us time step) is often referred to as "single-particle tracking" (See [6] to have more details.)
As the number of particles may be very large (hundreds to thousands), requiring "multiple-particle tracking" manual annotation of the image data is not feasible, and computer algorithms are needed to perform the task. At present, dozens of software tools are available for

particle tracking and several algorithms have been recently proposed aimed at solving this problem (see for example [32])
Nevertheless, the majority of the particle-locating algorithms are quite elaborated, and thus they turn out to be quite complex to be implemented and to yield a poor CPU-time performance. Furthermore, some particle-locating approaches are only valid for certain applications, such as two- dimensional (2D) grids or limited Eulerian cell displacements (see [7]).
In the previous F3MPIC particle tracking scheme a fast and simple priority-sorting algorithm was chosen to manage boundary crossing and particle tracking localization. The main effort of the development of F3MPIC was devoted to realize a robust and fast algorithm for tracking each particle inside the unstructured mesh.
Several different algorithms for the management of exception case have been implemented to avoid long searching loop. The topology of the mesh graph is obtained by means of an efficient front advancing scheme to allow a low computational cost. As a result, devices with shapes of arbitrary complexity can be easily treated, as for example imported from a 3D CAD model. The precautions that have been taken in the first phase of implementation have greatly reduced the number of loops searching needed to detect particle in an unstructured mesh; however algorithms working with advancing front are also easily prone to numerical errors due to the complexity of the geometry and these errors are difficult to predict and to categorize. For this reason it was necessary to substitute the old particle tracking presented in F3MPIC and to replace it with a new one and more flexible tracking method, that would meet the demanding just listed.
An efficient and robust particle-localization algorithm for unstructured grids was presented by Haselbacher A. and others (see [18]). Given a particle position and the cell containing the particle, the algorithm determines the cell which contains a nearby position. The algorithm is based on tracking a particle along its trajectory by computing the intersections of the trajectory and the cell faces. Compared to previously published particle-localization algorithms, the new algorithm has several advantages.
Firstly, it can be applied to grids consisting of arbitrary polyhedral cells. Secondly, the algorithm is not limited to small particle displacements. Thirdly, the interaction of particles with boundaries is dealt with correctly and naturally. Fourthly, the algorithm is more efficient than other published algorithms.
Starting from section 4.3 we will show this new algorithm in detail, while in the next section we will introduce a different test called *Particle or Point in tetrahedron test*. This test will be used for the validation of the new tracking algorithm. Finally, starting from section 4.4, we will also introduce a completely new charge deposition algorithm equipped with a secondary electron emission subroutine fully integrated with the new tracking method.

4.2 Particle in tetrahedron test

In this section we will introduce the easier to implement particle tracking algorithm. The great disadvantage of this scheme is connected with the fact that this algorithm is very hexose from a computational point of view, indeed it provides to search the chosen particle in all tetrahedra or generally in all mesh entities.
It is clear that, on very dense mesh i.e. high number of tetrahedra, an algorithm of this type is not efficient and for this reason it was necessary to find an easier method that allowed

us to determine if a given point, that corresponds to a given particle position, is located or not inside a given tetrahedron, once the coordinates of the vertices of the tetrahedron are known.
It is also important to stress that this test, called in future *particle or point in tetrahedron test*, is important not only because could be considered as the first form of tracking method used in an unstructured grid, but also because it will help us for the validation of the new, high efficient particle tracking code proposed by Haselbacher A. and others (see [18]). This one will be introduced in the next section.

Definition 2. *Let the tetrahedron have vertices*

$$V_1 = (x1, y1, z1) \tag{4.1}$$
$$V_2 = (x2, y2, z2) \tag{4.2}$$
$$V_3 = (x3, y3, z3) \tag{4.3}$$
$$V_4 = (x4, y4, z4) \tag{4.4}$$

and calling with P the test point (i.e. the point that we want to track) of coordinate

$$P = (x, y, z) \tag{4.5}$$

Then point P is in the tetrahedron if following five determinants all have the same sign.

$$D_0 = det \begin{bmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{bmatrix} \tag{4.6}$$

$$D_1 = det \begin{bmatrix} x & y & z & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{bmatrix} \tag{4.7}$$

$$D_2 = det \begin{bmatrix} x1 & y1 & z1 & 1 \\ x & y & z & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{bmatrix} \tag{4.8}$$

$$D_3 = det \begin{bmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x & y & z & 1 \\ x4 & y4 & z4 & 1 \end{bmatrix} \tag{4.9}$$

$$D_4 = det \begin{bmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x & y & z & 1 \end{bmatrix} \tag{4.10}$$

Proof. The proof is elementary and for our purpose is not important. See http://steve.hollasch.net/cgindex/geometry/ptintet.html to have more details. □

Some additional notes. Calling $i = 1, 2, 3$:

- If by chance $D_0 = 0$, then the tetrahedron is degenerate (the points are coplanar).
- If any other $D_i = 0$, then P lies on boundary i (boundary i is the boundary formed by the three points other than V_i).
- If the sign of any D_i differs from that of D_0 then P is outside boundary i.
- If the sign of any D_i equals that of D_0 then P is inside boundary i.
- It must be that $D_0 = D_1 + D_2 + D_3 + D_4$.
- The quantities $b_i = D_i/D_0$ are the barycentric coordinates.
- Comparing signs of D_i and D_0 is a check to test that P and V_i are on the same side of boundary i.

This test is very simple to implement and it will be used in what follows.
In the next section we will enter deeply in the new implemented algorithm and we will show its characteristics in great detail.

4.3 An efficient particle-localization algorithm for unstructured grids

The basic idea of the particle tracking algorithm presented by Haselbacher A. and others (see [18]) is the following: assume that the particle is known to be located in cell C_1 and it is known to move along a given trajectory.

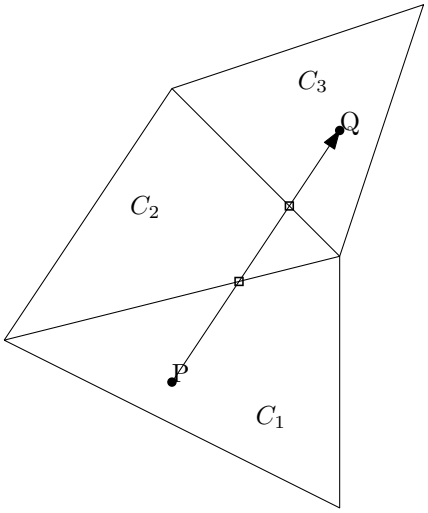


Fig. 4.1: *Illustration of particle-localization problem: find the cell which contains the particle position* $\mathbf{Q}$ *given that the particle was known to be located at* $\mathbf{P}$ *in cell* C_1

It is not important to know the shape of the trajectory that can be rectilinear as also parabolic and so on.
Assume further that it is possible to determine which face of cell C_1 is intersected by the particle trajectory. If the cell adjacent to the intersected face is C_2, the particle must pass from cell C_1 into cell C_2. By applying this idea repeatedly, we can determine the cell C_n which contains the predicted new particle position. In figure 4.1 this problem is represented in a clear way in fact given the particle position $\mathbf{r_P}$ and the cell which contains this position (in figure C_1), the aim is to find the cell which contains the nearby particle position $\mathbf{r_Q}$ (in figure C_3).
A cell is said to contain a particle location $\mathbf{r_P}$ if this position satisfies the so-called "in-cell test" i.e., for each face of the cell must holds that

$$(\mathbf{r_C} - \mathbf{r_p}) \cdot \mathbf{n} \geq 0 \tag{4.11}$$

where $\mathbf{r_C}$ is the centroid of the face and $\mathbf{n}$ is the outward unit normal of the face.
In our tetrahedral mesh $\mathbf{r_C}$ is easy to find in fact, given the coordinate of the tetrahedral vertices (x_i, y_i, z_i), with $i = 1, ...3$ of a given face, the face centroid is defined as

$$\mathbf{r_C} = (\frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3}, \frac{z_1 + z_2 + z_3}{3}) \tag{4.12}$$

Let's start now with the explanation of the new algorithm. Given the start point[1] $\mathbf{r_P} \equiv \mathbf{r_P}^n$ and the end point $\mathbf{r_Q} \equiv \mathbf{r_P}^{n+1}$, it is possible to compute the distance travelled by the particle during the time step defined as

$$d = \|\mathbf{r_P}^{n+1} - \mathbf{r_P}^n\| \tag{4.13}$$

It is also possible to find the direction versor defined as

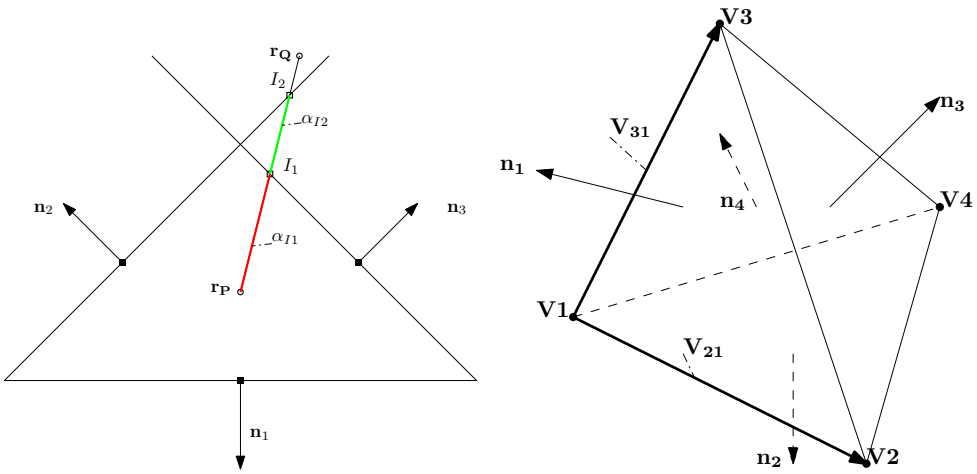
$$\mathbf{t} = \frac{\mathbf{r_P}^{n+1} - \mathbf{r_P}^n}{d} \tag{4.14}$$

[1] the index n is the time index

Let's start now considering only the cell which contains the particle position $\mathbf{r_P}$. We will also assume to work in a tetrahedral mesh like the one used in F3MPIC. The tetrahedral cell is defined by four vertices of coordinates in 3D: $\mathbf{V1}, \mathbf{V2}, \mathbf{V3}$ and $\mathbf{V4}$. In 2D the mesh is simply a triangular mesh, see for example figure 4.2. For clarity all the proposed figures except figure 4.2 b) are in 2D, but can easily extended in 3D.
The four vertices are connected to give faces and at each face we can assign an outward unit normal vector, let's call them $\mathbf{n_1}, \mathbf{n_2}, \mathbf{n_3}$ and $\mathbf{n_4}$. These vectors can easily be computed in the following way: once fixed a face, for example the face defined by the three vertices $\mathbf{V1}, \mathbf{V2}, \mathbf{V3}$ (see figure 4.2b)), it is possible to define the two position vectors $\mathbf{V}_{21} \equiv \mathbf{V2} - \mathbf{V1}$ and $\mathbf{V}_{31} \equiv \mathbf{V3} - \mathbf{V1}$. The normal vector to that face is defined as (calling it $\mathbf{n_1}$)

$$\mathbf{n_1} = \frac{\mathbf{V}_{21} \times \mathbf{V}_{31}}{\|\mathbf{V}_{21} \times \mathbf{V}_{31}\|} \tag{4.15}$$

Once the normal surface vector of this face is found, it is necessary to ensure that normal would be the outward normal i.e. its direction must be $\mathbf{n_1}$ or $-\mathbf{n_1}$[2].
To determine which normal, it is necessary to use the fourth (external) vertex of the tetrahedron: $\mathbf{V4}$.
Once chosen a vertex belonging to the face, for example $\mathbf{V1}$, it is possible to define the position vector $\mathbf{V_{41}} \equiv \mathbf{V4} - \mathbf{V1}$. In such a way the normal used is outward if the dot product $\mathbf{V_{41}} \cdot \mathbf{n_1}$ is negative. Other outward normals are computed in the same way.



(a)Intersection points

(b)Outward normal vectors. In this case we have considered a tetrahedral mesh element

Fig. 4.2: *In figure a) we have reproduced a graphical visualization of the intersection points using a triangular mesh element. In figure b) we have associated at each tetrahedral face its outward normal unit vector.*

Once the normal vectors at each face are computed, the algorithm computes the intersection points I_i of the trajectory with the faces. This aspect will be done in great detail in the next subsection. Now let's assume to have obtained a way to compute the intersection points I_i, and for clarity from now we will analyze the 2D case of figure 4.2 a), i.e. we will use a triangular mesh. The 3D case is the same, in fact from now a 2D or a 3D problem is the

[2]See http://math.stackexchange.com/questions/183030/given-a-tetrahedron-how-to-find-the-outward-surface-normals- for-each-side, to have more details.

same but the 3D problem is difficult to visualize using figures.
For each intersection point, associated to a specific side of the triangle of figure 4.2a), the associated intersection distance $\alpha_{I_i} = \|\mathbf{r_{I_i}} - \mathbf{r_P}\|$ is computed. However it is clear, from a computational point of view, how it is necessary to compute only the intersection points for which an intersection between the trajectory defined by $\mathbf{t}$ and the faces is possible i.e.

$$\mathbf{t} \cdot \mathbf{n} > 0 \tag{4.16}$$

It is important to note that, looking at figure 4.2a), some intersection points lie out of the triangles faces. However this problem is relative simple to solve; in fact it is necessary to choose, for what follows, only the intersection point with the smallest intersection distance. This is because, in traveling along the trajectory, the plane with the smallest intersection distance will be intersected first.
Once the algorithm determines which face is intersected by the trajectory finding the minimum of α_{I_i}, the particle can be assigned to the cell adjacent to that face, and the distance which remains to be travelled is updated according to

$$d \longleftarrow d - \min \alpha_{I_i} \tag{4.17}$$

After the particle is assigned to the new cell, the algorithm is simply applied again in the same manner until exceeds the distance which remains to be travelled.

4.3.1 Computation of trajectory-face intersections

In this section we will show how to compute the intersection points. The problem of finding the intersection of the particle trajectory with a planar face can be abstracted as determining the intersection of a ray $\mathbf{t}$ anchored at the point $\mathbf{r_p}$

$$\mathbf{r}(\alpha) = \mathbf{r_p} + \alpha\mathbf{t} \tag{4.18}$$

with a plane, specified by the normal vector $\mathbf{n}$, and anchored at the point $\mathbf{r_C}$

$$(\mathbf{r} - \mathbf{r_C}) \cdot \mathbf{n} = 0 \tag{4.19}$$

Substituting equation 4.19 into equation 4.18, we are able to obtain the distance between the intersection point $\mathbf{r_I}$ and $\mathbf{r_p}$ as

$$\alpha_I = \frac{(\mathbf{r_c} - \mathbf{r_p}) \cdot \mathbf{n}}{\mathbf{t} \cdot \mathbf{n}} \tag{4.20}$$

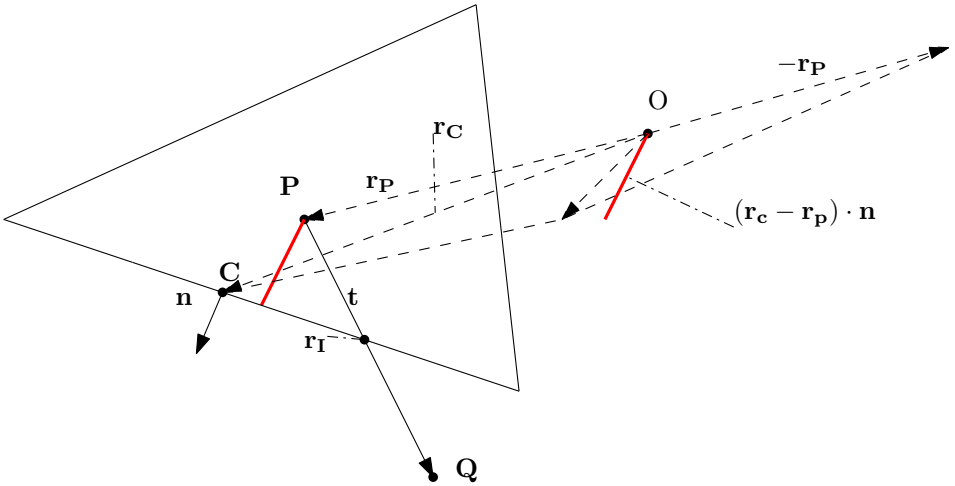


Fig. 4.3: *Geometrical construction used in the new tracking algorithm; the dot product* $(\mathbf{r_c} - \mathbf{r_p}) \cdot \mathbf{n}$ *in equation 4.20 is represented with a red line*

In figure 4.3 it is possible to appreciate the meaning of the various pieces of equation 4.20. In fact the numerator is the signed normal distance between the plane and the particle position; it is positive if the particle is located in the cell. The denominator indicates the orientation of the trajectory relative to the face normal; if the denominator is positive (negative), the particle is moving toward (away from) the face, and if it is zero, no intersection is possible. We will no enter in other computational discussion, in the next section we will only report a pseudo-code implementation and we refer the reader to [18] to have more details.

4.3.2 Pseudo-code formulation and implementation hints

In this section we will show the pseudocode of the algorithm explained in the previous section.
In particular following notation of [18], we call this algorithm *fast intersection algorithm*. In [18] there is also a different version of the previous algorithm called *robust intersection algorithm*. The last one is able to work with machine precision to avoid some computational problems that could occur if some strange particles loader is used (for example if particles are generated on a surface of the thruster used as a planar nozzle). However, in F3MPIC, the loader works directly with tetrahedra: particles are loaded in tetrahedra and for this reason the *robust intersection algorithm* isn't necessary.
Programming the present algorithm is relatively simple. The only data structures required are face-to-cell and cell-to-face connectivity tables. In F3MPIC to each tetrahedron are associated four faces and four nodes. At each face, called f in the following pseudocode, are associated two pointers. These two pointers point directly at the two tetrahedra shared by the face. The face with the minimum intersection distance, i.e. the face which therefore corresponds to $\alpha_{min} \equiv \min \alpha_{I_i}$, is called f_{min}. If the face is a boundary face, one of these two pointers is set to the NULL pointer during mesh loading; in this way it is quite easy to see if the intersected face is a boundary face. It is sufficient to check how many NOT NULL pointers are found given a f_{min} face. If this number is one, the face is a boundary face, otherwise the face is an internal face.
For reference, the present particle-localization algorithm is summarized below. The two

functions BOUNDARY DEPOSITION($c_P, f_{min}, \mathbf{r_p}$) and INTERNAL DEPOSITION(c_P^{n+1}) are connected with charge deposition and will be explained in the following section.

Algorithm 4 (c_P^{n+1}) = PARTICLE TRACKING($\mathbf{r_P}^{n+1}, \mathbf{r_P}, c_P^n$)

$d = \|\mathbf{r_P}^{n+1} - \mathbf{r_P}^n\|$
$\mathbf{t} = \frac{\mathbf{r_P}^{n+1} - \mathbf{r_P}^n}{d}$
$\mathbf{r_P} = \mathbf{r_P}^n$
$c_P = c_P^n$
while $d > 0$ **do**
    (α_{min}, f_{min}) = INTERSECTION($\mathbf{t}, \mathbf{r_P}, c_P$)
    $\mathbf{r_P} \longleftarrow \mathbf{r_P} + \alpha_{\mathbf{min}}\mathbf{t}$
    $d \longleftarrow d - \alpha_{min}$
    if $d \geq 0$ **then**
        if f_{min} has not a **NULL** pointer **then**
            c_P = number of the tet. at which belongs the face ($\neq$ previous c_p)
        else
            f_{min} has a **NULL** pointer $\longrightarrow$
            $\longrightarrow$ BOUNDARY DEPOSITION($c_P, f_{min}, \mathbf{r_p}$)
            Break or nothing (it depends from boundary type)
        end if
    else
        $c_P^{n+1} = c_P$
        Gather function to find weights on tetrahedra
        INTERNAL DEPOSITION(c_P^{n+1})
        Break
    end if
end while

The most important part of the previous pseudocode is the subroutine used to compute the intersection point, called (α_{min}, f_{min}) = INTERSECTION($\mathbf{t}, \mathbf{r_P}, c_P$).
In this case, following what explained in the previous section, it is quite easy to find the intersections point, in fact it is sufficient to compute the normal vector at each face and then to compute α_I as explained in equation 4.20.
In what follows we summarize the pseudocode used for this function.

Algorithm 5 (α_{min}, f_{min}) = INTERSECTION($\mathbf{t}, \mathbf{r_P}, c_P$)

$\alpha_{min} = \infty$
for all $f \in$ [tetrahedron in which the particle is located] **do**
    $\mathbf{n} = \mathbf{n}(f)$ Find the normal to that face
    if $\mathbf{t} \cdot \mathbf{n} > 0$ **then**
        $\mathbf{r_c} = \mathbf{r_c}(f)$
        $\alpha_I = \frac{[(\mathbf{r_c} - \mathbf{r_p}) \cdot \mathbf{n}]}{\mathbf{t} \cdot \mathbf{n}}$
        if $\alpha_I < \alpha_{min}$ **then**
            $\alpha_{min} = \alpha_I$
            $f_{min} = f$
        end if
    end if
end for

It is clear that, to obtain a completely particles localization, it is necessary to apply this algorithm for all the particles in the domain and, in principle, if the particle density would be hight, the time necessary to compute the intersection points could grows quickly.
However, in standard PIC formulation, particles can't move more of a tetrahedron at each time step to fulfill Debye's compatibility and it is clear that, if this happens, the algorithm becomes very efficient.
The only thing that remains to explain is the new charge deposition; this will be done in the following section. In particular in this new code we have also introduced a completely new algorithm to manage plasma-surface interaction with secondary electron emission.

4.4 Boundary management and charge deposition

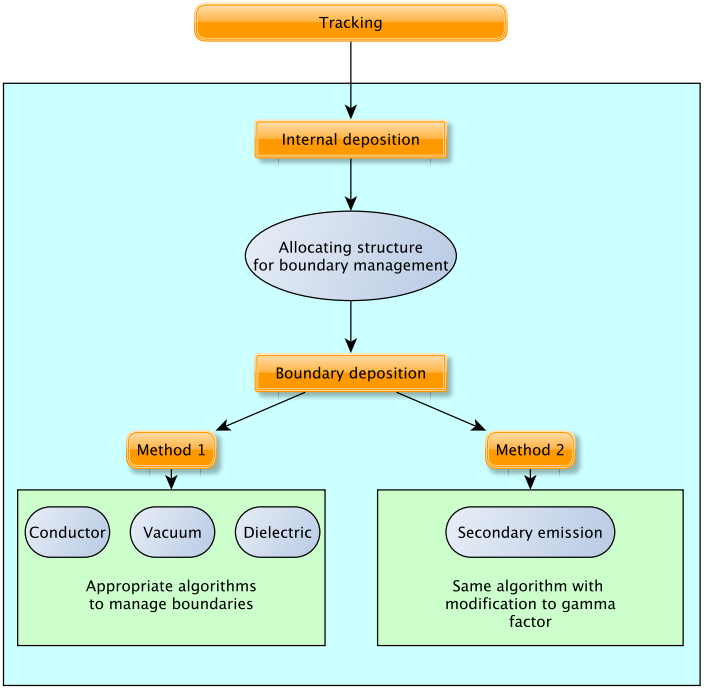


Fig. 4.4: *In figure we have reproduced the actual structure of F3MPIC. The user can choose how to perform boundary deposition. In particular there are two different ways to manage boundaries called Method 1 and Method 2. Method 1 is a revival of the method previously available in F3MPIC; Method 2 is instead a completely new way to manage boundaries in which a strong emphasis has be done to the subroutine that manages the emission of secondary electrons*

In this section we show the main characteristics of the new charge deposition. Let's start with one of its associated function: INTERNAL DEPOSITION(c_P^{n+1}).
This function is used only for particles that do not exit from the domain at a given time-step, as shown in the pseudocode of the previous section.
Once found the new tetrahedron in which the particle is located, this function computes the charge density on tetrahedral mesh nodes.
We will not enter into detail in the particular weighting scheme used to compute charge

density because, for our purpose, is not particularly interesting. An example of internal deposition performed using a structured mesh can be found in chapter 6.
Let's start now to analyze the management of the boundaries.
In this new implementation there are two different ways of treating the interaction of the particles with the walls, called in the following Method 1 and Method 2. The user can choose the implementation that prefers, see figure 4.4.
In both implementations at each wall is assigned a structure; this structure contains all the information connected to the boundary, for example its material, its threshold energy, its output streams i.e. the number of particles exited from the wall in a given time-step and so on.
Regardless of the method chosen, the boundary management is always arranged with the BOUNDARY DEPOSITION $(c_P, f_{min}, \mathbf{r_p})$ function, but this function is radically different in the two implementations.
Method 1 is a modification of the previous F3MPIC boundary management to integrate it in the new particle tracking algorithm; in particular each type of boundary is treated in a different manner and at each type of boundary are associated precise numbers and algorithms to manage physically the behavior of the particles at the wall.
The second, and completely new implementation: Method 2 is radically different; in fact each type of wall is conceptually treated in the same way. What differentiates the behavior of the particles at a given wall is related to the secondary emission coefficients γ which characterizes the wall itself i.e. the material of which the wall is made. In particular this new method is thought to consider the emission of secondary electrons from the wall due to the interaction with the plasma. These two methods will be analyzed in the following sections.

4.4.1 Method 1: Boundary management

Let's start with the boundary management performed with Method 1.
In this method at each wall are associated two numbers. The first one is called the **Physical Entity** number, while the second one is called the **Boundary type** number. The **Physical Entity** number could be considered as the identify number of the wall while the **Boundary type** number is used instead to characterize wall material. If the first number could be any integer number (i.e. it depends on GMSH), the second number is assigned as proposed in the following table

Tab. 4.1: Type of boundaries

Number	Type of boundary
-1	CONDUCTOR
-2	VACUUM
> 0	DIELECTRIC

As shown in the previous table, we have introduced three kinds of boundaries: conductors, dielectrics and the so called vacuum boundaries. In real life there are other important boundaries, like for example emitters that can be considered as boundaries and so on. However, for our practical purpose, these types of boundaries are enough. Vacuum boundaries are

boundaries to control streams. They aren't real physical boundaries; they are only used to save quantities.
Let's start now with the description of the management of conductor boundaries.
Conductors are materials that permit electrons to flow freely. A wall made of a conducting material will permit charge to be transferred across the entire surface of the object. If charge is transferred to the wall at a given location, that charge is quickly distributed across the entire surface of the wall. Since conductors allow for electrons to be transported from particle to particle, a charged object will always distribute its charge until the overall repulsive force between excess electrons is minimized. Generally the wall is put to ground. For this reason once a charge hits the conductor, it flows to the ground and it is neutralized. This is equivalent to discharge the particle once it hits the face f that belongs to a conductor boundary. In our unstructured mesh these faces are triangles and are known. In fact with the new tracking algorithm we know exactly where the particle exits from the domain. For this reason it is quite simple to treat conductive walls, once the particle goes out from a conductor face, i.e. a face that " has a **NULL** pointer", the particle is simply discharged.
Dielectric walls are treated differently, in fact a dielectric material is an electrical insulator that can be polarized by an applied electric field. In [18] dielectric deposition is just mentioned; for this reason it is interesting to show how we have solved the problem of dielectric deposition.
If a particle hits a dielectric has two possible scenarios: if the particle is reflected, it is necessary to track its new position, considering that the particle could have changed the tetrahedron in which the reflection has been performed; otherwise, if it is absorbed by the boundary, a local deposition has to be performed. We will now analyze these two cases, leaving the end of this section to discuss when to perform a reflection and when an absorption.
Let's start with the first case: particle reflection, look at figure 4.5

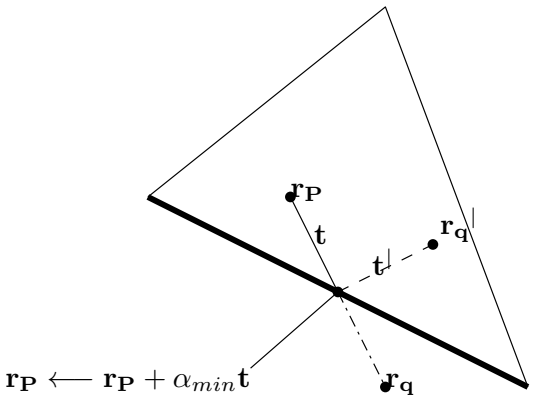


Fig. 4.5: *In figure we have reproduced a geometrical construction to analyze particle reflection. The point in which the particle hits the wall is known thanks to the new tracking algorithm. To perform the reflection it is only necessary to compute* $\mathbf{t}'$ *and* $\mathbf{r_{q'}}$

The first step is to compute the new particle versor (due to particle reflection) trying to find the new particle position after the reflection i.e. $\mathbf{r_{q'}}$. This can be done in an easy way using

appropriate transformations; in fact it is sufficient to reflect the previous final point $\mathbf{r_q}$ (that is located outside mesh boundaries) using the following transformation to find $\mathbf{r_{q'}}$

$$\mathbf{r_{q'}} = \mathbf{A}\mathbf{r_q} + \mathbf{b} \tag{4.21}$$

where the precise forms of $\mathbf{A}$ and $\mathbf{b}$ depend on the boundary condition. For example, for a solid or symmetric boundary, that is our case, $\mathbf{A} = \mathbf{I} - 2\mathbf{n}\mathbf{n^T}$ where $\mathbf{n^T}$ and $\mathbf{n}$ are the transpose and the outward normal vectors to the face, $\mathbf{I}$ is the identity matrix and $\mathbf{b}$ is $\mathbf{b} = 2(\mathbf{r_C} \cdot \mathbf{n})\mathbf{n}$. Once the reflected position is known, it is necessary to compute the intersection point between the previous trajectory $\mathbf{t}$ and the plane at with the intersection point belongs. With the tracking algorithm of the previous section the intersection point is known, in fact is $\mathbf{r_P}$ itself,[3] so the new trajectory versor can be found as follows

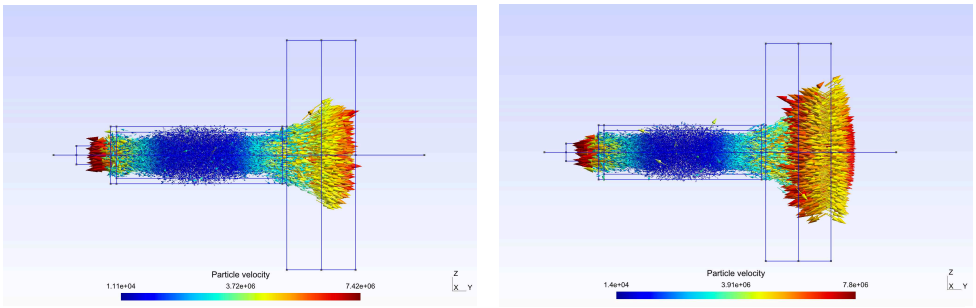
$$\mathbf{t'} = \frac{\mathbf{r'_q} - \mathbf{r_P}}{d'} \tag{4.22}$$

where $d' = \|\mathbf{r'_q} - \mathbf{r_P}\|$.
Computationally, the boundary deposition with dielectric reflection could be summarized with the following pseudocode. See also figure 4.6 for a visual example of the new algorithm.

Algorithm 6 BOUNDARY DEPOSITION$(c_P, f_{min}, \mathbf{r_P})$: DIELECTRIC REFLECTION

```
if REFLECTION=TRUE then
    Compute r'_q with 4.21
    Compute d' = ||r'_q − r_P||
    Compute t' as 4.22
    d ⟵ d'
    t ⟵ t'
end if
```

If the particle is not reflected, it is deposited exactly in the point in with it hits the wall i.e. $\mathbf{r_P}$. In such a way the charge contributes to the surface charge density of the wall.



(a)Electron velocities before collision with a dielectric boundary

(b)Electron velocities after collision with a dielectric boundary

Fig. 4.6: *In figure we have shown the effect of particles collisions with a dielectric boundary using Method 1; in particular, in these figures, it is possible to appreciate electron velocities before and after the collision. In figure b) it is evident the reverse direction of the electron speeds.*

[3] After the advancing i.e. $\mathbf{r_P} \longleftarrow \mathbf{r_P} + \alpha_{\mathbf{min}}\mathbf{t}$

The last thing to conclude this section is to show how to choose when a particle is reflected and when a particle is deposited on the wall. In this implementation the particle is deposited or reflected in accordance with two quantities: the reflection probability and the threshold energy of the wall. These parameters are strictly related to wall material and they determinate the behavior of the particle on the wall.

4.4.2 Method 2: Boundary management

Let's start with the description of the boundary management performed with Method 2.
As in Method 1, also in Method 2 we use the **Physical Entity** number to identify the wall but, in this case, different types of walls are not divided using the **Boundary type** number. In fact each type of wall is conceptually treated in the same way; what differentiates the behavior of the particles at a given wall is essentially related to the secondary emission coefficients. In fact, in this new model, when a particle hits a boundary, regardless of the type of material, a number of secondary electrons are generated at the surface. The number of produced electrons is managed with the secondary electron emission coefficient called γ, defined as the number of electrons emitted per incident electron.
In other words, if we call I_s the electron flux that leaves the surface and I_0 the electron flux that strikes the surface, we can write γ as

$$\gamma = \frac{I_s}{I_0} \tag{4.23}$$

Following the implementation proposed in [36], this coefficient is dependent on the electron primary energy and on the polar angle relative to the surface normal i.e. the angle formed by the velocity vector of the primary electron that strikes the surface and the normal of the surface itself. Depending on the value of gamma, which contains all the information necessary to characterize the wall, it is possible to manage walls made of different materials i.e. conductors and insulators.
In this implementation, when a particle (electron or ion) strikes the wall, it leaves the simulation. If the particle is an electron it could induce an emission of secondary electrons characterized by a secondary electron emission coefficient γ.
Each of the produced electrons have an energy that follows an experimentally secondary electron distribution function.
Baroody ([3]) observed that, irrespective of the material, the general shape of this curve, as a function of the primary electron energy and of the angle of incidence measured relative to the surface normal, tends to a universal curve if normalized to the maximum value of γ called γ_{max}. In particular the γ_{max} value occurs at a particular energy that in the following we will define as E_{max}.
Generally γ_{max} is approximately 1 in metals at a primary energy E_{max} of several hundred eV. Values of γ_{max} and E_{max} in insulators are typically much higher (see figure 4.8a)). In figure 4.9b) are given typical values of γ taken from [35].
A good mathematical expression to compute γ has been derived by Schwarz ([34])

$$\gamma(E_{pe}, \theta) = 2.6\gamma_{max}\epsilon^{\frac{2}{3}}G_n(\epsilon, \theta) \tag{4.24}$$

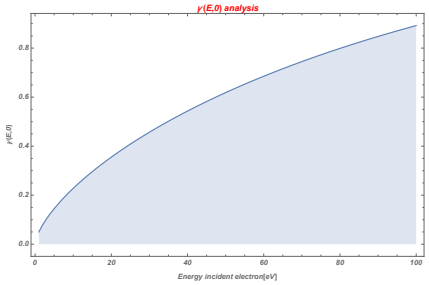
where ϵ is

$$\epsilon = 0.72 \frac{E_{pe}}{E_{max}} \tag{4.25}$$

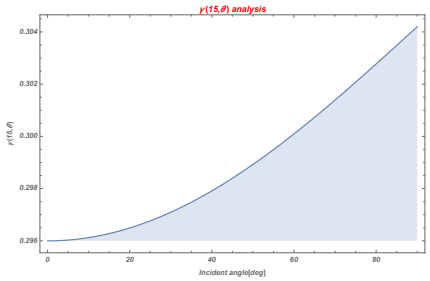
and $G_n(\epsilon, \theta)$ is defined as

$$G_n(\epsilon, \theta) = \frac{1 + 2\epsilon \cos^{1/(n-1/3)}(\theta)}{(1 + \epsilon \cos^{1/(n-1/3)}(\theta))^3} \tag{4.26}$$

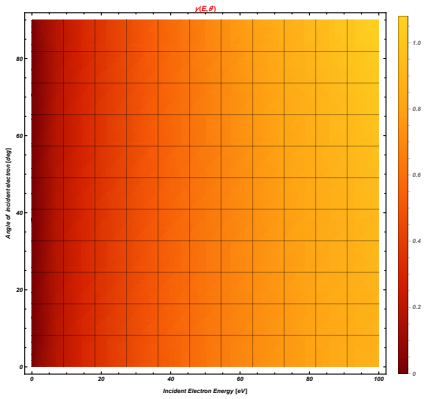
n is the so called angular exponent, and it is in the range $n \in [\frac{4}{3}, \frac{5}{3}]$, E_{pe} is the energy of the primary electron ($\equiv$ pe) that has induced a secondary emission and θ is the angle formed by the direction of the incidence electron and the surface normal.



(a)Analysis of $\gamma(E_{pe}, \theta)$ as a function of energy of the incidence particle keeping fixed $\theta = 0$



(b)Analysis of $\gamma(E_{pe}, \theta)$ as a function of the angle of incidence keeping fixed $E_{pe} = 15eV$



(c)Density plot to analyze $\gamma(E_{pe}, \theta)$

Fig. 4.7: *In figure we have analyzed the secondary emission coefficient γ defined in equation 4.24. In this case the analysis has be done for iron in which $\gamma_{max} = 1.3$ and $E_{max} = 400eV$. In figure a) we have reproduced the trend of the factor γ as function of the energy of the incident electron. In particular we have worked in conditions of normal incidence i.e. $\theta = 0$. In figure b) we have analyzed the dependence of the factor γ from the angle θ, keeping the primary electron energy fixed at $E_{pe} = 15eV$. The plot c) is a simple density plot varying both the quantities.*

In the following we will block n to $4/3$, in fact in [36] it is also shown as γ depends very little to n. In figure 4.7 we have reproduced the γ factor for the iron. In this case $\gamma_{max} = 1.3$ and $E_{max} = 400eV$. As shown in figure 4.7a), in normal incidence conditions, already at $20eV$, γ is ~ 0.4 and this aspect simply shows as secondary electron emission in a plasma is not negligible considering that, generally, the electron thermal agitation energy is $\sim 15eV$. In figure 4.7c) we have also shown a density plot.
When the factor $\gamma(E_{pe}, \theta)$ is known, we use it to test if an emission occurs, precisely every unit in the γ factor produces a secondary electron emission, while its fractional part is

statistically tested to see if it also contributes to the production of secondary electrons. For example, if we get a γ factor of 1.2, it is firstly produced a secondary electron and secondly the remaining part of the γ factor, i.e. 0.2, it is used to test the probability that a new emission can occur. In other words a random number r in the interval $[0, 1]$ is generated and, if this number is less than 0.2, a new secondary electron is emitted. Each produced electron follows a typical energy distribution. In figure 4.8b) we have reproduced an example of the used distribution.

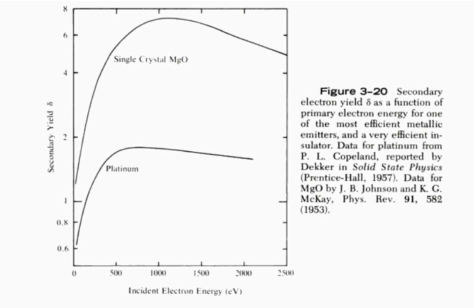
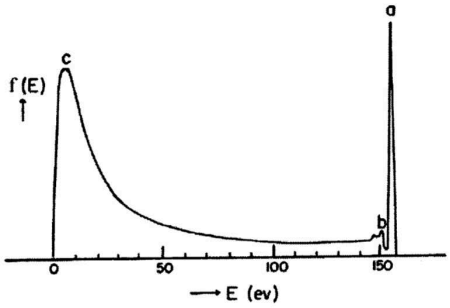


Figure 3-20 Secondary electron yield δ as a function of primary electron energy for one of the most efficient metallic emitters, and a very efficient insulator. Data for platinum from P. L. Copeland, reported by Dekker in *Solid State Physics* (Prentice-Hall, 1957). Data for MgO by J. B. Johnson and K. G. McKay, Phys. Rev. **91**, 582 (1953).

(a)Secondary electron emission coefficient. In figure caption there is the used reference



(b)Distribution function of emitted electrons.

Fig. 4.8: *In figure we have reproduced two historical charts to analyze the secondary emission coefficients. In particular in figure a) it is clear how the number of emitted electrons for insulator is generally greater of the number of emitted electrons for conductors. In figure b) there is a real experimental distribution function for secondary emitted electrons used to model the simulative pdf in equation 4.27*

According to figure 4.8b) the energy of the produced electron can be divided in different groups.

- The peak indicated by $a)$ corresponds to elastically reflected primaries.
- The peak indicated by $b)$ are electrons emitted with energies between $E = 50eV$ and the primary energy; they have suffered discrete energy loss and are then referred to as: inelastically reflected primaries
- The bulk of emitted electrons that have low energy correspond to the peak indicated by c). These are referred to as: true secondary electrons, implying that they are electrons which originally occupied bound states in the crystal structure

Since the fraction of elastically and inelastically reflected primaries is usually small, the term "secondaries" will be assumed to include all emitted electrons. In particular according to [8], the shape of the energy distribution of the emitted secondaries i.e. E_{se}, is essentially independent of the energy of the primary electron (for primary electron energy below than keV) and it is given by[4]

$$f_{E_{se}} \equiv f(E_{se}) = \frac{6E_{se}\chi^2}{(E_{se}+\chi)^4} \tag{4.27}$$

[4]normalized in interval $[0, +\infty]$

where χ is the surface binding energy (work function for metals and electron affinity for insulator), see figure 4.9a).

Ag	4.26 – 4.74	Al	4.06 – 4.26	As	3.75
Au	5.1 – 5.47	B	~4.45	Ba	2.52 – 2.7
Be	4.98	Bi	4.31	C	~5
Ca	2.87	Cd	4.08	Ce	2.9
Co	5	Cr	4.5	Cs	2.1
Cu	4.53 – 5.10	Eu	2.5	Fe:	4.67 – 4.81
Ga	4.32	Gd	2.90	Hf	3.9
Hg	4.475	In	4.09	Ir	5.00 – 5.67
K	2.29	La	3.5	Li	2.9
Lu	~3.3	Mg	3.66	Mn	4.1
Mo	4.36 – 4.95	Na	2.36	Nb	3.95 – 4.87
Nd	3.2	Ni	5.04 – 5.35	Os	5.93
Pb	4.25	Pd	5.22 – 5.6	Pt	5.12 – 5.93
Rb	2.261	Re	4.72	Rh	4.98
Ru	4.71	Sb	4.55 – 4.7	Sc	3.5
Se	5.9	Si	4.60 – 4.85	Sm	2.7
Sn	4.42	Sr	~2.59	Ta	4.00 – 4.80
Tb	3.00	Te	4.95	Th	3.4
Ti	4.33	Tl	~3.84	U	3.63 – 3.90
V	4.3	W	4.32 – 5.22	Y	3.1
Yb	2.60 [13]	Zn	3.63 – 4.9	Zr	4.05

(a)Work function values i.e. χ for some elements, in units of electron volt.

Z	Atom	δ^m	E^m_{PE} (eV)	η ($E_{PE} > 10$keV)	δ (20 keV) calc.
3	Li	0.5–0.6	100–200	0.07	0.13
4	Be	0.5–0.9	200–300	0.08	0.18
5	B	1.0–1.2	150–400	0.08	0.34
6	C	0.9–1.0	300–1100	0.1	0.28
11	Na	0.65	300	0.19	0.17
12	Mg	0.8–0.9	300	0.2	0.24
13	Al	0.9–1.0	250–300	0.2	0.28
14	Si	0.9–1.1	250–300	0.22	0.30
15	P	(1.2)	(550)	0.23	0.38
16	S	(1.2)	(560)	0.25	0.38
19	K	0.5–0.7	300	0.28	0.15
20	Ca	(0.85)	(390)	0.28	0.24
21	Sc	0.8	300	0.29	0.22
22	Ti	0.7–0.9	300	0.3	0.22
23	V	(1.0)	(470)	0.31	0.30
24	Cr	(1.0)	(480)	0.32	0.30
25	Mn	(1.0)	(500)	0.32	0.30
26	Fe	1.1–1.3	400	0.33	0.38
27	Co	0.9–1.2	400–600	0.33	0.32
28	Ni	1.0–1.3	500–550	0.34	0.37
29	Cu	1.1–1.3	500–600	0.34	0.38
30	Zn	0.9–1.1	200–500	0.35	0.30
31	Ga	1.3	300–500	0.35	0.42
32	Ge	1.0–1.2	300–500	0.35	0.34
33	As	(1.4)	(670)	0.36	0.47
34	Se	0.6–1.3	400–500	0.36	0.28
37	Rb	0.6–0.9	350–400	0.37	0.20
38	Sr	0.8	250	0.37	0.22
39	Y	0.8	350–400	0.38	0.22
40	Zr	0.9–1.1	350	0.38	0.30
41	Nb	1.1–1.2	550	0.38	0.36
42	Mo	1.0–1.2	400	0.38	0.34
43	Tc	(1.1)	(560)	0.38	0.34
44	Ru	(1.1)	(580)	0.39	0.34
45	Rh	(1.1)	(580)	0.39	0.34
46	Pd	1.3	650	0.39	0.42
47	Ag	1.2–1.4	700–800	0.39	0.42
48	Cd	0.9–1.1	400–500	0.4	0.3
49	In	1.3–1.4	500	0.4	0.45
50	Sn	1.1–1.4	500	0.4	0.40
51	Sb	1.2–1.3	600	0.4	0.40
52	Te	1.4	700	0.41	0.47
55	Cs	0.5–0.8	300–400	0.41	0.17
56	Ba	0.7–0.9	400	0.41	0.22
57	La	0.8	500	0.41	0.22
58	Ce	(0.6)	(240)	0.41	0.15
72	Hf	1.1	460	0.43	0.34
73	Ta	1.0–1.35	600	0.43	0.37
74	W	1.0–1.4	700	0.43	0.38
75	Re	1.3	900	0.43	0.42
76	Os	1.3	(730)	0.43	0.42
77	Ir	(1.5)	(770)	0.43	0.52
78	Pt	1.35–1.7	700–750	0.43	0.52
79	Au	1.2–1.6	700–875	0.45	0.47
80	Hg	0.9–1.1	600	0.45	0.30
81	Tl	1.4	650	0.45	0.47
82	Pb	1.0–1.3	500–700	0.45	0.37
83	Bi	1.2–1.3	500–700	0.45	0.40
90	Th	0.9–1.3	600–800	0.45	0.34

(b)γ_{max} and E_{max} values for some elements

Fig. 4.9: *In figure a) we have shown the values of the work function χ for some conductive elements. The table has been taken from https://en.wikipedia.org/wiki/Work_function. In figure b) there are some values for γ_{max} and E_{max} for some typical conductors. This table has been taken from [35]. If compared with the notation used in this chapter the notation in [35], and used for this table, is slightly different.*

We have used distribution 4.27 to find the energy of the emitted particles. It follows that it is necessary to find a method to generate a random variable (in this case the energy of the emitted secondary electron i.e. E_{se}) with the just mentioned distribution.
Generally calling X the random variable of which we want to generate values, our objective can then be summarized as follow:

- Let X be a random variable whose probability density function is defined as f_X
- We want to generate values of X which are distributed according to this distribution.

This result can be obtained using the method called "Inverse transform sampling". This method is a basic method for pseudo-random number sampling, i.e. for generating sample numbers at random from any probability distribution given its cumulative distribution function F_X.
This method can be summed up using the following definition

Definition 3. *If X is a continuous random variable with cumulative distribution function F_X defined as $F_X(x) = \int_{-\infty}^{x} f_X(t)dt$, then the random variable $Y = F_X(X)$ has a uniform distribution on [0, 1] and the random variable $F_X^{-1}(Y)$ has the same distribution as X.*

Using this definition it is possible to obtain a practical guide to generate values of X which are distributed according to the desidered distribution. This guide can be summarized as follows:

1. Generate a random number u from the standard uniform distribution in the interval $[0, 1]$.

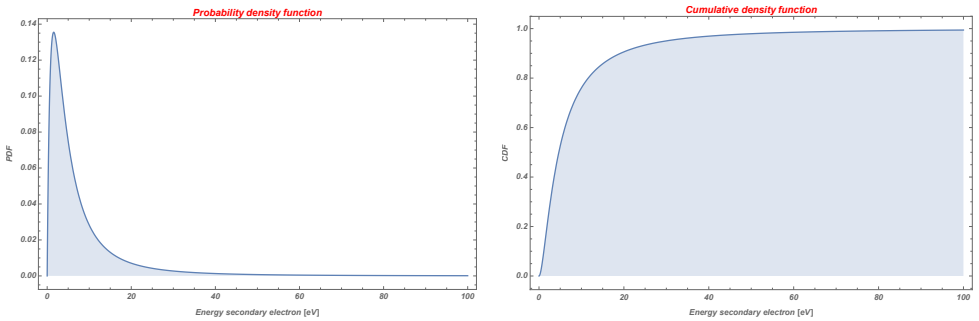
2. Compute the value $x \in X$ such that $F_X(x) = u$ i.e find x for which $x = F_X^{-1}(u)$

3. The x value just obtained follows the probability density function f_X

Therefore, if we have a random number generator to generate numbers according to the uniform distribution, and if we are able to find an analytic expression for the cumulative density function F_X, associate to the probability density function f_X, we can generate any random variable with a known distribution.
In our case the random variable X is E_{se}, with probability density function $f(E_{se})$ defined in 4.27. Fortunately the problem of generating a uniform distribution in the range $[0, 1]$ can be easily solved using the C function $drand48()$. Also the problem to find an analytic expression for the cumulative density function $F_{E_{se}}$ associated to $f(E_{se})$ can be easily solved in fact

$$F_{E_{se}}(e_{se}) = \int_0^{e_{se}} f_{E_{se}}(t)dt = 6\chi^2\left(\frac{1}{6\chi^2} - \frac{\chi + 3e_{se}}{6(\chi + e_{se})^3}\right) \tag{4.28}$$

where E_{se} is defined only for positive values and e_{se} is a value assumed by the variable E_{se}. The main problem of the strategy defined in the previous itemize, is the point 2). In fact to find x, it is necessary to find F_X^{-1} and it is not said that F_X is invertible (This is the case if we consider $F_{E_{se}}$). The problem to find F_X^{-1} however can be easily solved by referring to the above problem in a different way. In fact we do not focus on the pretense of finding F_X^{-1} but we want only to find solutions even approximate of the equation $F_X(x) = u$ or, in other words, we want to find the roots of the function $g(x)$ defined as $g(x) \equiv F_X(x) - u$ i.e $g(x) = 0$. This problem can be solved computationally using some root-finding methods, for example bisection method, in fact the function F_X is strictly increasing.
Applying the whole method just described, we can generate values e_{se} that follow the probability density function defined in 4.27.
In figure 4.10 we have reproduced the probability density function $f(E_{se})$ and its cumulative density function $F_{E_{se}}$.



(a)Probability density function *(b)Cumulative density function*

Fig. 4.10: *In figure we have reproduced the probability and the cumulative density function associated to the random variable E_{se}. The probability density function has been defined in 4.27, while the cumulative in 4.28. In both cases the analysis has be done for iron*

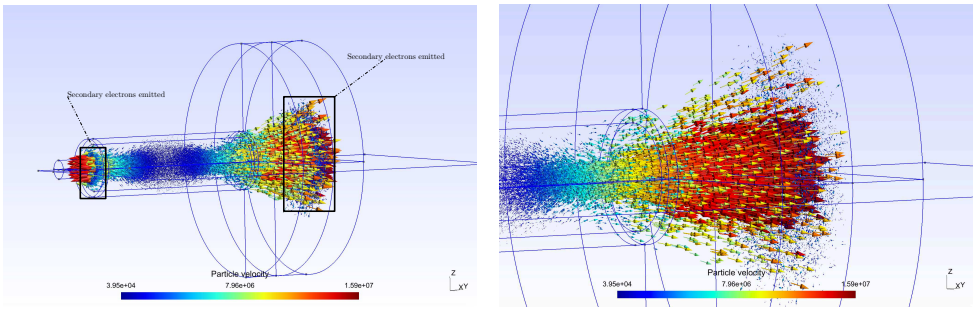
Once the energy of the emitted electron i.e. the sampled e_{se} is found with the bisection method (in our case with an error <1%), the new particle is generated in the exact position in which the primary electron hits the surface. The velocity, in module, of the emitted electron can be easily found from its energy. The only thing that remains to explain is the angular distribution of the emitted electron i.e. the versor of the emitted velocity. This one is assumed to be uniform in the azimuthal plane (ϕ) and cosine ($\cos(\theta)$) in the polar angle relative to the surface normal. This type of distribution can be easily found in the following way

- Generate a random number ϕ uniform distributed in the interval $[0, 2\pi]$
- Generate a random number $\nu \equiv \cos(\theta)$ uniform distributed in the interval $[-1, 1]$
- Derive the angle θ as $\theta = \arccos[\nu]$

The velocity versor of the emitted electrons: $\hat{\mathbf{v}}$, can be found as follows

$$\begin{aligned} \hat{v_x} &= \sin(\theta) * \cos(\phi) \\ \hat{v_y} &= \sin(\theta) * \sin(\phi) \\ \hat{v_z} &= \cos(\theta) \end{aligned} \tag{4.29}$$

However the velocity versor $\hat{\mathbf{v}}$ must only be generated inside the boundary limit i.e. inside the chamber. To check if the generated versor is in the right direction, we make the dot product of $\hat{\mathbf{v}}$, with the outward normal to the surface: $\mathbf{n}$. If the dot product is positive, all signs of versor $\hat{\mathbf{v}}$ must be reversed.
The entire procedure is repeated for all electrons that hit a wall. The obtained result is shown in figure 4.11. In this case we have increased the potential difference between the electrodes and reduced plasma density in order to increase electron speed and therefore the energy of the electrons that impact on the wall; in this way we maximize the production of secondary electrons.



(a)Graphical visualization to appreciate the production of secondary electrons

(b)Zoom of figure a)

Fig. 4.11: *Graphical visualization to appreciate the production of secondary electrons, in particular the two surfaces located at the top and at the bottom of the device are involved in the production of secondary electrons. Figure b) is a zoom of figure a) to better appreciate secondary electron production*

4.5 Validation of the new code and comparison with the old one

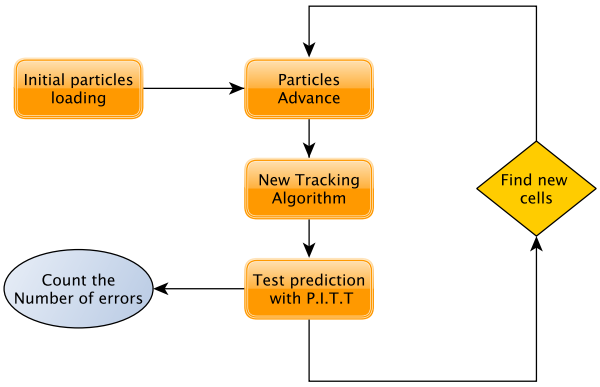


Fig. 4.12: *Scheme of the cycle used to test the new tracking algorithm. The index prediction, obtained with the new algorithm, is tested with the P.I.T.T. (point in tetrahedron test, see section 4.2) algorithm and if the test fails a counter is updated*

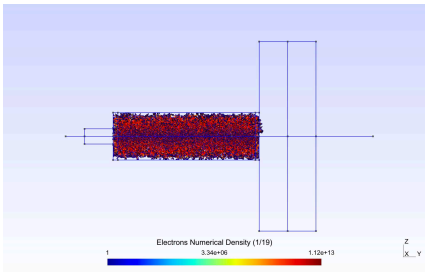
In the previous section we have shown the main characteristics of the new track particle algorithm with a particular attention to the new charge deposition algorithm.
A first strong validation of the new tracking algorithm has be done using the "particle in tetrahedron" test explained in section 4.2.
In figure 4.12 we have shown the main cycle used to test the new algorithm. We have performed different simulations loading a different number of particles in the system and following each particle. We have then tested the prediction of the tetrahedron index, obtained using the new tracking algorithm, with the "particle in tetrahedron" test. If the test failed, a counter was updated. We don't have obtained error messages using this test and this is enough to validate the new tracking algorithm.
In addition to test the localization algorithm, it was also necessary to test the new charge

deposition algorithm. To do this we have done different simulations using the same experimental setup and the same experimental conditions described in chapter 3.
Differently form the previous chapter, in this case we are interested in two main aspects

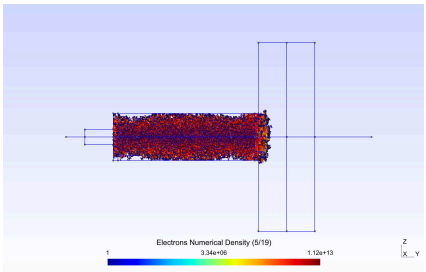
1. Looking at the evolution of charge density or equivalently looking at the evolution of the numerical density of each species in the system (as clear numerical density and charge density are closely connected)

2. Looking at particles flux at each wall, i.e. exit rate at each wall, and since our main aim is to characterize a plasma thruster, to use these fluxes to obtain a detailed analysis of the performance of the engine with a particular focus to the specific impulse of the thruster

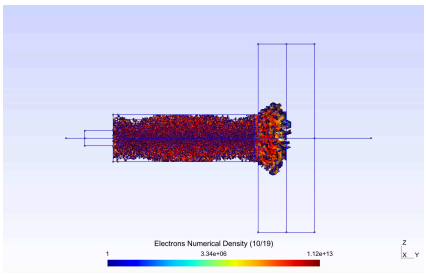
Let's start with the evolution of the numerical density i.e. point 1). The electrons and ions velocities have been already shown in chapter 3 in figure 3.9.
As stated, if a particle remains in the domain, it is deposited. To do this, as explained in previous section, it is necessary to weight its charge on the tetrahedral mesh nodes in which the particle is located. This weighting is done with the new INTERNAL DEPOSITION(c_P^{n+1}) function.
In figure 4.13 we have shown the electron numerical density evolution following the system for twenty global time-steps.



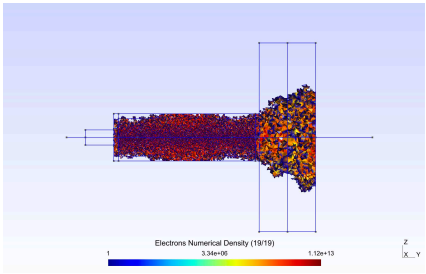
(a)Total numerical electron density at time step number: 1



(b)Total numerical electron density at time step number: 5



(c)Total numerical electron density at time step number: 10



(d)Total numerical electron density at time step number: 20

Fig. 4.13: *In figure we have plotted the evolution of electron density. At the beginning electrons are localized in the source region; then they follow the electric field and at the end they exit from the bottom of the device.*

As clear, the new algorithm works well; electrons change their position and the electron numerical density changes in time following electron movements.
In particular the evolution of the electron density is fully compatible with the electric field

configuration of figure 3.8.
Let's start now to analyze particles flux i.e. point 2) of the previous itemize. Thanks to new structure allocated to save informations on boundaries, at the end of each time-step an output file is always proposed as shown in the following table.

Tab. 4.2: *Flux analysis. In particular at each wall are associated: its **Physical Entity** number, its **Physical Type** number, the number of particles exited from that wall and the associated flux*

Physical Entity	Physical Type	Electrons exited	Exit Rate
610	-1	1263	$1.263e^{12}$
609	-1	34	$3.4e^{10}$
447	-1	12	$1.2e^{10}$
448	-1	2	$2e^{9}$
608	-1	7	$7e^{9}$
607	-1	47	$4.7e^{10}$
606	-1	0	0
309	-1	132	$1.32e^{11}$
308	-1	907	$9.07e^{11}$
299	-1	226	$2.26e^{11}$

The first two columns are the **Physical Entity** number and the **Physical Type** number of the geometry presented in the previous chapter.
In this case the **Physical Type** number is always -1 because in this implementation all the walls are conductive walls. A quick visualization of the **Physical Entities** numbers is shown in figure 4.14 where, near each wall, we have associated its **Physical Entity** number.
The third column is the number of electrons exited from each surface during the time-step (in this implementation $1ns$), and the last column is the exit electron rate computed as

$$\text{Rate} = \frac{\text{Number of particles exited}}{\text{Time step lenght}} \tag{4.30}$$

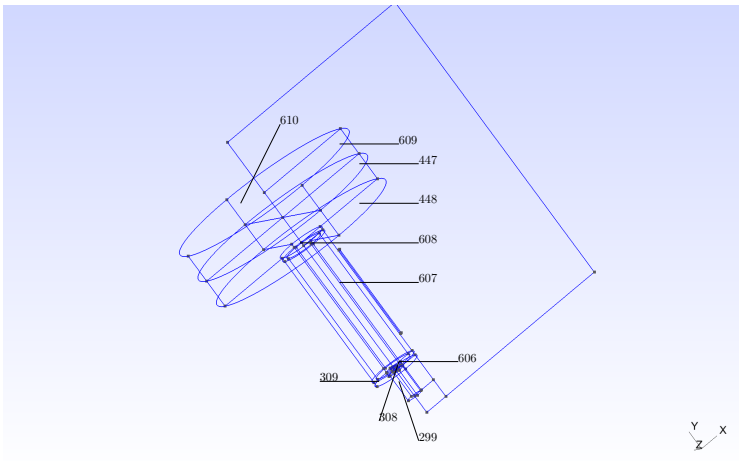


Fig. 4.14: *Representation of the physical entities numbers of table 4.2. The used geometry is the same of the previous chapter*

To measure the efficiency of the thruster the first parameter to compute is the specific impulse, called $I_{sp/\mathrm{TOT}}$.
The dominant contribution is given by the electrons with their high speeds and, if we call $< v_e >$ the mean electron velocity measured at a fixed wall along the axis of the engine, the total specific impulse can be estimated as follows

$$I_{sp/\mathrm{TOT}}[\mathrm{s}] \sim \frac{< v_e >}{g_0} \tag{4.31}$$

The thrust of the thruster is instead defined as

$$S[\mathrm{N}] = g_0 I_{sp/\mathrm{TOT}} \frac{dm_{\mathrm{TOT}}}{dt} \sim g_0 I_{sp/\mathrm{TOT}} \frac{\Delta m_{\mathrm{TOT}}}{\Delta t} \tag{4.32}$$

If electrons, due to their smaller mass and high velocity contribute mainly to $I_{sp/\mathrm{TOT}}$, ions and neutral particles contribute mainly to S. Performing a simulation using the new developed tools and looking at the total $I_{sp/\mathrm{TOT}}$ in the physical entity placed at the bottom of the system, we have obtained the following result

$$I_{sp/\mathrm{TOT}} = 10^5 \text{ s} \tag{4.33}$$

Actually neutral particle movement in F3MPIC is not implemented, and also the density used for this simulation is relatively low, for this reason an estimate of S loses meaning. The obtained $I_{sp/\mathrm{TOT}}$ is very high but, looking at figure 3.9 and at the estimate of electron velocity in section 3.4, it is fully reasonable.
A first analysis of this value could be done looking at figure 4.15.

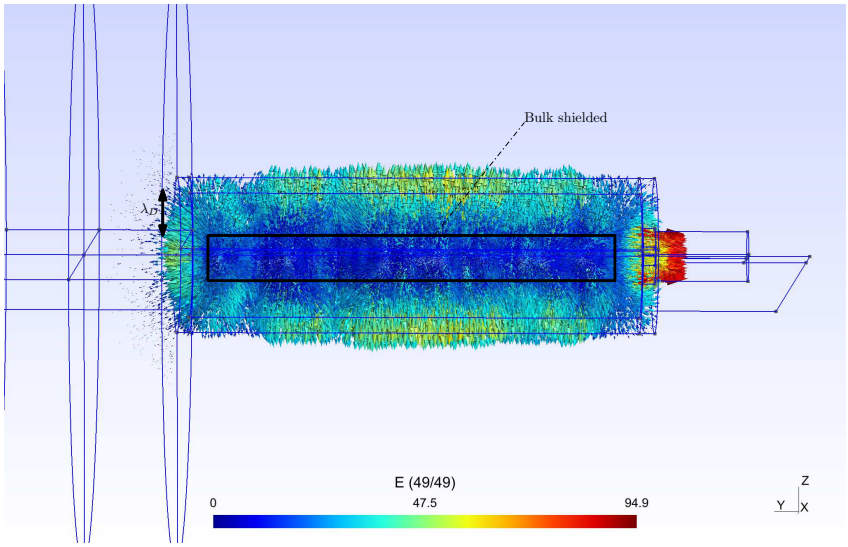


Fig. 4.15: *In figure we show the electric field configuration without an external electric field. In particular we have underlined the Debye length λ_D. If we consider the radius r of the source as a characteristic dimension of the system it is clear as the condition $r >> \lambda_D$ it is not strictly fulfilled. However the plasma bulk is quite shielded*

In figure 4.15 we have switched off the potential difference and we have analyzed only the electric fields due to particle motion. It is clear as the plasma bulk is quite shielded from electric fields. However the shielding effects could be better, this is related with the

compatibility with Debye's length showed in the same figure.
In these simulation electron temperature was $11600K$ and electron density was of $10^{11}m^{-3}$.
Using these informations and remembering the definition of Debye's length as

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{N_e q_e^2}} \tag{4.34}$$

we can obtain an estimate of the Debye's length for this configuration: $\lambda_D \sim 0,02m$.
The radius of the discharge region is of $r = 0,025m$. From this should be clear as the shielding effect perpendicular to the y axis, which is the axis of symmetry of the system, it is not very effective and this is the main responsible of the high specific impulse.
In fact if we add an external electric field, as the one explained in the previous chapter, electrons are accelerated to very high speeds ($v_e \sim 10^6 m/s$), without being affected by the almost-neutrality of the plasma and the condition $r >> \lambda_D$, that is necessary to produce plasma, it is not strictly fulfilled.
These problems in line of principle could be easily solved growing computational density but, for a code that runs on single processor, this means to significantly increase the computational time.
To test this hypothesis we have increased particle density of an order of magnitude increasing plasma rate production, and we have done the same analysis just explained. In this new configuration $\lambda_D \sim 0,007m$, that is lower if compared with the previous λ_D. In particular the obtained results show as the global electric field is lower if compared with the previous one and the plasma bulk region is greater.
Starting from this condition, we have recomputed the specific impulse, obtaining:

$$I_{sp/\text{TOT}} = 10^3/10^4 \text{ s} \tag{4.35}$$

If compared with previous $I_{sp/\text{TOT}}$, the new value is more realistic. However this value continues to be high and difficult to achieve in practice. The reason is linked mainly because the starting conditions for these simulations are just rough conditions; in particular the starting condition for plasma density is pretty rough.
For this reason these estimates are only indicative, and a more accurate analysis should be done after have calibrated the simulation using real experimental data.

4.5.1 Efficiency of the new model

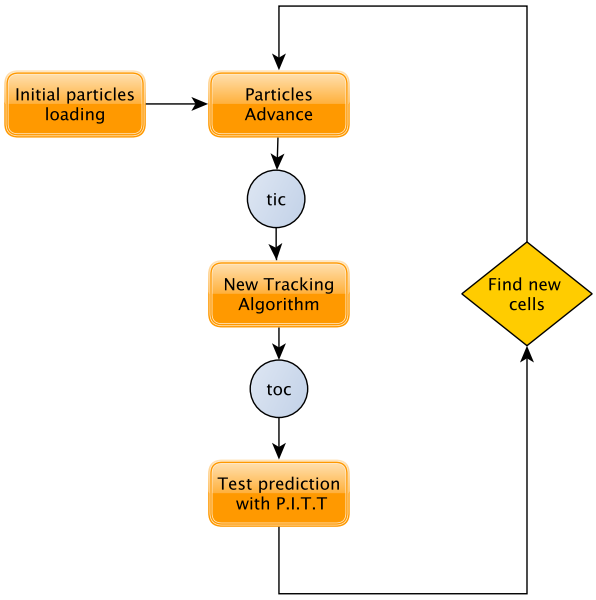


Fig. 4.16: *Scheme of the cycle used to test the new algorithm and to compute its efficiency. Note the two variables that it is necessary to compute at each cycle called* ***clock_t tic = clock()*** *and* ***clock_t toc = clock()****. These ones are used to obtain an estimate of the time necessary to perform a completely particles localization.*

Until now we were only interested to test the "physics" of the new algorithm. It's now time to see its efficiency when it is compared with the other two particles tracking methods introduced in the previous sections: the point in tetrahedron test and the old F3MPIC particle tracking.
With efficiency we refer to the time required by the algorithm to accomplish a complete particle localization i.e. to find all the new cells in which particles are located, varying the number of the particles in the system.
This time can be obtained quite easily with a small peace of code inserted directly before "to call" the function that is necessary to use to perform particles localization.
This code is a simple C library function called **clock()** that returns the number of clock ticks elapsed since the program was launched.
To obtain a time it is necessary to call this function twice: before and after the particle tracking function is called. See figure 4.16
To get the number of seconds used by the CPU to execute function, it is only necessary to divide by **CLOCKS_PER_SEC** the difference between the two output computed by the clock function.
On a 32 bit system, where **CLOCKS_PER_SEC** equals 1000000, this function returns the same value approximately every 72 minutes. However this number depends strongly on the machine. See next listing for clarity. [5]

[5] See for reference http://www.tutorialspoint.com/c_standard_library/c_function_clock.htm

```
#include <stdio.h>
#include <time.h>

        ........

clock_t tic = clock(); //First calling

//Type of tracking
Point_in_tet() || New_track_part() || Old_F3MPIC_track_part();

clock_t toc = clock(); //Second calling

//Difference
printf("Elapsed: %f seconds\n", (double)(toc - tic) / CLOCKS_PER_SEC);

        ........
```

Using this function we were able to estimate the execution time of these algorithms. In particular, fixed a tracking localization algorithm, and fixed the number of particles in the system ($50000, 500000$ and so on..), we have performed for each group of loaded particles four independent simulations and we have then controlled the time necessary to execute a completely particles localization. Let's call these times t_1, t_2, t_3, t_4.
To obtain a realistic value, we have computed the mean time necessary to track particles with a given particle localization algorithm i.e. $\hat{t} = \frac{\sum_{i=1}^{4} t_i}{4}$ and we have associated to it the usual mean error defined as $\frac{\sigma}{\sqrt{4}}$, where σ is the usual standard deviation. The obtained results are presented in figure 4.17 and in the next tables.

Particles tr.	$\hat{t}[s]$	**Error[s]**
50000	0,355	0,005
500000	2,375	0,005
5000000	24,915	1,115
10000000	48	0,12

Param.	**Value**	**Error**
m	$3,26 * 10^{-6}$	$1,74 * 10^{-8}$
q[s]	$-0,05$	$0,09$

Tab. 4.3: *Estimate of execution time for the "old F3MPIC particle tracking algorithm"*

Particles tr.	$\hat{t}[s]$	**Error[s]**
50000	1,39	0,02
500000	13,33	0,01
5000000	133,21	0,04
10000000	267	0,915

Param.	**Value**	**Error**
m	$2,67 * 10^{-5}$	$2,29 * 10^{-8}$
q[s]	$-0,03$	$0,121$

Tab. 4.4: *Estimate of execution time for the "point in tetrahedron test"*

Particles tr.	$\hat{t}[s]$	Error[s]
50000	0,205	0,005
500000	1,445	0,005
5000000	16,365	2,565
10000000	32,535	5,165

Param.	Value	Error
m	$4.81 * 10^{-6}$	$6.75 * 10^{-8}$
q[s]	0.21	0.37

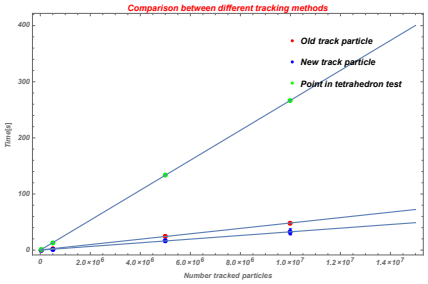
Tab. 4.5: *Estimate of execution time for the "new particle tracking algorithm"*

In previous tables we have also reported the results of the fits showed in figure 4.17 a). In particular we have interpolated data with a linear function of the type $y = mx + q$, where $x \equiv$ Number tracked particles, and $y \equiv$ Time.
It is easy to see that the linear trend is well tested for all three tracking methods with a percentage error on m of about 0.01%; this fact allows us to conclude that the time necessary to track particles in the system increases linearly with the number of particles.
To test this linearity, we have performed some simulations changing the density of tetrahedra[6] and in all cases the linear trend was well occurred. In particular we have noticed that the only parameter, as expected, that is affected by the complexity of the mesh is the slope of the straight line in fact, increasing tetrahedra density, we also increase the slope, increasing in this way the time required by the algorithms to track particles

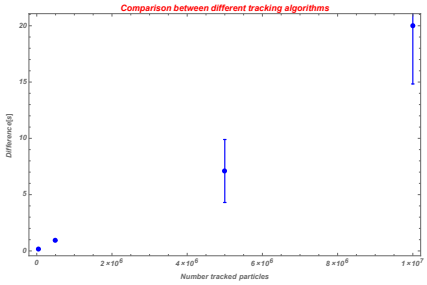
$$m \propto \text{Tetrahedra density} \tag{4.36}$$

We conclude this section giving a practical estimate of the different efficiency of the three algorithms.
Looking at figure 4.17a), it is clear as the new tracking algorithm is more efficient if compared with the previous ones; in particular this efficiency goes from 20% to 50%. Generally it is difficult to determine in advance how much better (in percentage) is the new algorithm compared with previous ones, because each test is bound by particles position that is not replicable changing simulation. However similar simulations have yielded a result comparable to that shown in figure 4.17c) and, for this, we conclude that the new algorithm is on average more efficient of 40% when compared with the previous ones.
As expected, the slower track particle method is the "*point in tetrahedron test*", explained at the beginning of this chapter that, as already stated, was only used to test the prediction of the new track particle algorithm.

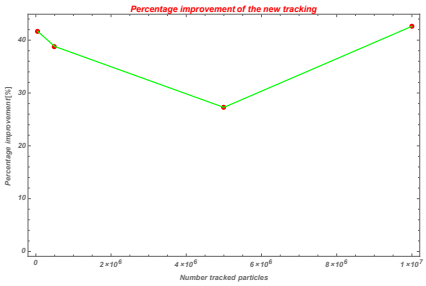
[6] In the new algorithm the tetrahedra density is directly connected with the number of intersection points that it is necessary to compute



(a)Comparison between different particles tracking methods



(b)Difference in time between the old track particle algorithm and the new track particle algorithm



(c)Percentage improvement between new track particle algorithm and old track particle algorithm

Fig. 4.17: *Comparison between different tracking algorithms used in F3MPIC. In figure a) we have plotted the time that each algorithm takes to perform a completely particle localization in function of the loaded particles. The slower tracking method is the "point in tetrahedron test" marked with green dots, while the faster is the new track particle algorithm marked with blue dots. The red dots represent the execution time of the old F3MPIC tracking algorithm. In all cases the time necessary to track particles in the system increases linearly with the number of loaded particles. In figure b) it is possible to appreciate the difference in time between the old track particle algorithm and the new one, in particular in figure c) is represented the percentage improvement of the new algorithm if compared with the old one.*

4.6 Conclusions

In this chapter we have shown a new particle tracking localization algorithm. After introducing a first basic tracking algorithm called *point in tetrahedron test()*, the new tracking algorithm has been explained in detail. In particular the prediction of the new tracking method has been deeply tested with the *point in tetrahedron test()*.
If compared with previous F3MPIC tracking algorithm, we have shown how the new algorithm is faster and more efficient. This efficiency goes from 20% to 50% and it depends strongly from the complexity of the mesh and not easily a priori predictable. We have also shown a new algorithm to manage boundaries and internal deposition directly integrated inside the new tracking algorithm. In particular different types of boundaries have been managed and the deposition algorithm has been extensively tested.
The integration of charge deposition inside tracking could be considered a novelty in the PIC standard where generally the tracking and deposition phases are split. The novelty of the new deposition system is also closely linked to the introduction of a new model to manage secondary electrons emission.

Simulation results

5

> *There are children playing in the streets who could solve some of my top problems in physics, because they have modes of sensory perception that I lost long ago.*
>
> **— J. Robert Oppenheimer**

Abstract

In the previous chapters we have introduced some new algorithms to manage the interactions between particles. All these algorithms have been tested using well established theoretical models. As explained in the introduction of this thesis, in its original implementation F3MPIC was developed for the detailed design and optimization of helicon and general-purpose plasma thruster and it has been validated both numerically and experimentally under the HPH.COM. The aim of this chapter is to use the new implemented algorithms to test a new type of HPT thruster that is now in development at CISAS, a group of Padua university.
Nowadays, with the current implementation of F3MPIC, an exhaustive characterization of the source is not yet feasible. In fact the real experimental plasma density is very high, i.e. $\sim 10^{19} m^{-3}$, and it is difficult to replicate keeping sustainable computing times. This is a direct consequence of the fact that F3MPIC, in its current implementation, runs on a single processor. For this reason we have decided to analyze only the involved electromagnetic fields. In the months to follow, we will complete the parallelization of F3MPIC and the new code will be used to conclude the analysis started in this chapter.

5.1 Introduction

The Helicon Plasma Thruster (HPT) has been presented during the last decade as a novel electric propulsion device, see [5].
The HPT is composed of the following parts (see figure 5.1)

- A cylindrical chamber, where plasma is produced, typically made of dielectric material, i.e., Pyrex glass
- A radio-frequency (RF) antenna wrapped around the chamber, that emits within the range 1-50MHz, with a wide assortment of topologies: annular, Nagoya-III type, helical. See figure 5.2

- A RF subsystem necessary to power the antenna. This consists of a power unit, a wave generator/amplifier and a matching network, which adapts the RF power to the plasma electromagnetic behavior

- A feeding system that is commonly attached to the back of the chamber

Finally, a set of several electromagnets and/or permanent magnets, surrounding the chamber, generate the required magnetic field inside the chamber (mainly axial) and in the plasma expansion area, forming a divergent magnetic nozzle (MN) topology.

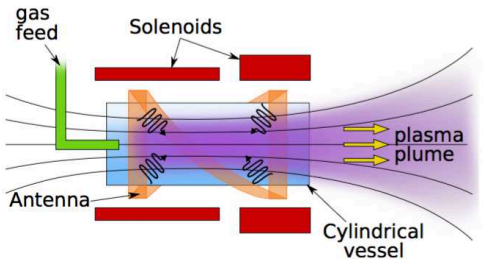


Fig. 5.1: *Sketch of the HPT with the main parts. See [5]*

Different physical processes take place in a HPT, involving among others: the emission and the propagation of the waves from the antenna to the plasma and the absorption of the RF waves energy, which is deposited mainly on the electrons. In particular the energized electrons bombard the neutral gas, producing a high density plasma.
The generated plasma is confined and guided by the magnetic field; a forward acceleration of ion is driven by the ambipolar electric field which naturally develops within the plasma to sustain quasi-neutrality.
Thrust is understood as the increment of the momentum of the supersonic beam. The produced thrust is delivered to the thruster thanks to the interaction of plasma currents with the applied magnetic field.
The attractiveness of these devices in comparison with other electric propulsion devices, such as Hall thrusters, ion engines or MPDs, is connected with the fact that this type of thruster does not need any electrode, grids or neutralizers. The lack of these components suggests that the HPT is a simple and robust device.
A long lifetime is also expected since the limited plasma-wall interaction, due to the magnetic confinement, reduces contamination or sputtering of sensitive components, e.g. the cathode in Ion or Hall thrusters. The beginnings of the studies in Helicon Sources is attributed mainly to R. Boswell and Chen

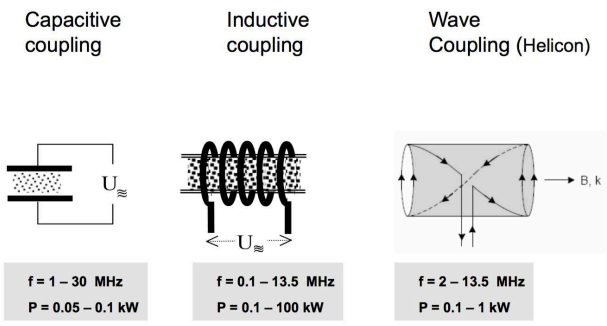


Fig. 5.2: *Examples of RF antennas, assortment of different topologies. See [25]*

HPT prototypes are often classified according to the magnetic circuit they use and the power range in which they operate. Several research groups have developed HPTs that implement permanent magnets, mostly in the low power range, below 1kW. The Permanent Magnet Expanding Plasma (PEMP) built at the University of Tokyo, the Helicon Plasma Hydrazine COmbined Micro (HPH.COM) funded by the European 7th FrameWork Program, and the Compact Helicon Plasma Thruster, designed at the Institute of Nuclear Research of the Ukranian National Academy of Sciences, are some examples of HPTs that use permanent magnets to generate the magnetic field.
Recently a new project called SAPERE has started. The aim of this project is to project and to test a high-power ($>$ 1KW) plasma thruster. The consortium for the design and the realization of SAPERE is led by Thales Alenia Space Italia and involves also CISAS, a group of Padua University, as one of the major partners.
SAPERE is organized in two sub-projects: STRONG and SAFE. SAPERE/STRONG aims at the realization of a reusable space tug coupled to the rocket VEGA for the transfer of payloads of different sizes from an intermediate orbit to the target orbit.
In this chapter we will use the new numerical tools, developed in previous chapters, to begin the characterization of a new RF thruster prototype (see figure 5.3b)). This prototype is slightly different from the first STRONG prototype (see figure 5.3a)), but it has essentially the same plasma generating mechanisms. In particular the obtained results for this smaller device will be used, in the following months, as test benchmark and with a scale up mechanism they will help us in the optimization of the existing STRONG prototype.

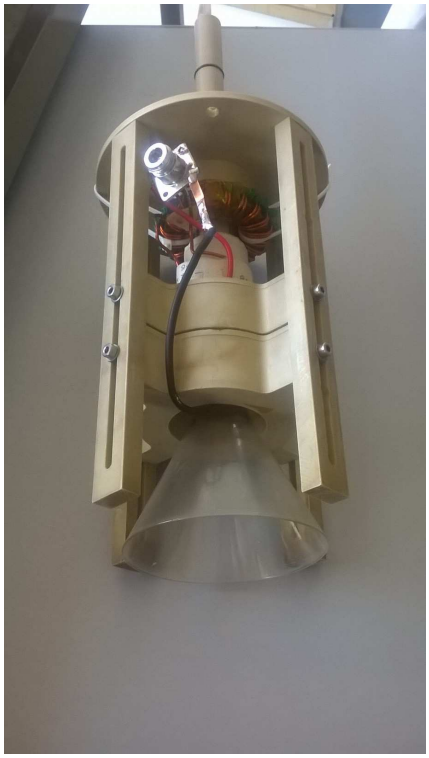
5.2 Experimental setup

In the previous section we have introduced the characteristic in common to all HPTs thrusters. In particular we have introduced the plasma production mechanism and the involved physics. These characteristics are also at the basis of the HPT thruster prototype in development at CISAS, that is reproduced, for clarity, in figure 5.3b). This thruster is an HPT thruster with a desired thrust T of $T \sim 500/600 \mu N$.
The entire device weighs less than $1kg (< 300gr)$ and the used gas is Argon. The amount of used gas is in the range $\sim [0.02 - 0.1]$ mgr/s.
The material with which the engine has been developed is hexagonal boron nitride. The design and the materials of the thruster are still subject of studies, and they were constantly

evolved, during the first design stage, in an effort to find the sweet spot between weight, propulsion and cost.
This device will be mounted on board of a micro-satellite. Due to the extreme environments that satellites must endure, the material that has been chosen for this thruster required a very demanding set of properties, involving among others:

- High dielectric strength. The material must be a very good dielectric due the hight voltages involved
- Thermal shock resistant. Generally satellites are exposed to extreme temperature fluctuations in space. It follows that the chosen material must have a good thermal shock resistant
- High thermal conductivity. This need follows because generated electrical conditions require adequate cooling
- Low coefficient of thermal expansion. Due to temperature fluctuations it is imperative that the thruster and the micro-satellite remain dimensionally stable
- Low density. The weight of space components is critical for efficient launching and other operations

(a)Prototype developed during the STRONG project



(b)Prototype of the new HPT thruster

Fig. 5.3: *Left: Prototype developed during the STRONG project. Right: Prototype of the new HPT thruster that has been simulated in this chapter. In this figure you can appreciate the connector to power the antenna*

The hexagonal boron nitride is a material that has consistently met the requirements of dielectric strength, thermal shock resistance, machinability and secondary-ion erosion resistance and, for this reason, it has been chosen as the material for this new device.
In addition to the highly innovative material, one of the main characteristics of this HPT thruster or, more generally, one of the peculiarities of all CISAS plasma projects, is the use of a novel type of antenna to produce plasma, called S-Helicon antenna.
In figure 5.3b) it is possible to appreciate the connector to power the antenna.
The idea at the basis of this type of technology, is the following: to maximize the power deposited by the RF antenna into neutral gas and so to maximize the amount of produced plasma, it is necessary to resonate with one of the typical plasma oscillation frequencies. For a reason that will become clear in the following, the frequency with which it would be necessary to resonate is the cyclotron frequency.
To explain in detail thes last statement, it is necessary to analyze the involved plasma frequencies. In first approximation, it is possible to consider a plasma as a harmonic oscillator with different oscillation frequencies. Among these we find, the plasma frequency

$$\omega_P[\text{rad/s}] = \sqrt{\frac{N_e q_e^2}{\epsilon_o m_e}} \tag{5.1}$$

due to plasma oscillation to restore quasi-neutrality; and the cyclotron frequency, due to the presence of an external magnetic field

$$\omega_c = -\frac{q\mathbf{B}}{m} \rightarrow \omega_c[\text{rad/s}] \equiv |\omega_c| = \frac{q|\mathbf{B}|}{m} \tag{5.2}$$

where q is the charge of the analyzed particle. Their associated frequencies are $f_i \sim \frac{\omega_i}{2\pi}$ with $i = P, c$. These ones have the following order of magnitude

$$f_P \sim \text{ GHz} \tag{5.3}$$

and

$$f_C \sim \text{ MHz} \tag{5.4}$$

The plasma frequency varies too quickly and therefore it is not possible to resonate with it; it follows that the only frequency with which it is possible to resonance is the cyclotron frequency.
As stated at the beginning of this chapter, the HPTs thrusters work with a radio-frequency (RF) antenna wrapped around the chamber that emits, generally, within the range 1-50MHz. From the last discussion should be now clear why the chosen working range is precisely this, in fact, in this way, the external frequency is of the same order of magnitude of the cyclotron frequency.
If it were possible to obtain a coupling with the cyclotron frequency, the advantages would be considerable. In fact it is possible to demonstrate that the power deposited in the plasma by a RF antenna, is function of the external frequency and of the oscillating plasma frequencies in the following way

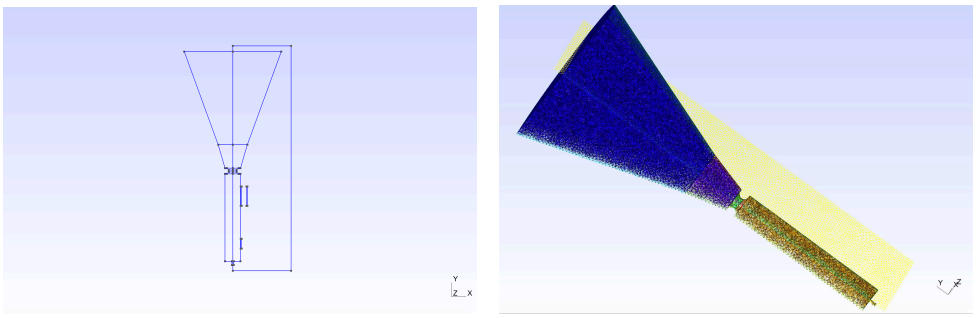
$$P_{\text{abs}}[\text{Watt}/m^3] = N_e \frac{e^2 E_0^2}{2m_e}\left(\frac{\nu}{\nu^2 + f_i^2}\right) \tag{5.5}$$

where ν is the external frequency, f_i is one of the characteristics plasma oscillation frequencies and N_e is the numerical electron density.

From this equation follows that P_{abs} reaches its maximum when $\nu \sim f_i$.
Another innovative feature of Cisas thruster, compared with other ones, is the use of a diaphragm at the chamber exit, which constricts the flow, increasing the density within the chamber, and consequently improving ionization.
Different arrangements of permanent magnets are currently in development in order to optimize the magnetic topology. The field strength is in the range 400-1100G, but the non-uniformity of the topology makes difficult the understanding of all the involved phenomena including wave propagation and plasma flow behavior.
The same "aspect" of the thruster is not yet defined, and a more detailed study of the different dimensions is necessary. For all these reasons, in order to optimize the current prototype, it is necessary to perform several simulations at the purpose to find the best configuration. In the next section we will focus on the analysis of the electromagnetic fields. The real characterization of the new source will be performed only when the new F3MPIC implementation, based on a MPI parallelization, will be ready.

5.3 RF discharge: Preliminary analysis of the new Cisas thruster prototype

In this section we will show some of the preliminary results that have been obtained for the electromagnetic fields.
In figure 5.4 we have reproduced, using GMSH, the new thruster. In particular, in figure 5.4a), it is possible to appreciate its geometry, reproduced using real dimensions, while in figure 5.4b) we have represented the used mesh.



(a)GMSH view of the simulated geometry *(b)GMSH view of the used mesh*

Fig. 5.4: *Left: GMSH view of the simulated geometry for the new HPT thruster. Right: GMSH view of the used mesh*

Let's start with the analysis of the electric field.
The first step to obtain useful results is to find a way to consistently simulate the time-varying electric fields due to the presence of the external radio-frequency antenna mounted on the dielectric tube inside which there is the neutral gas. The RF antenna, in figure 5.4a), is represented like little rectangles.

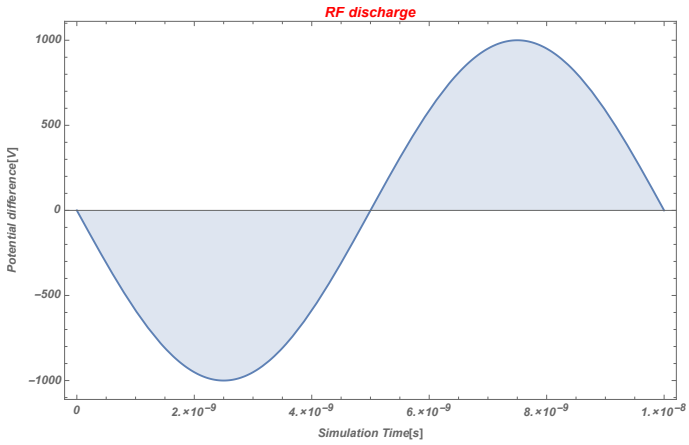


Fig. 5.5: *In this figure we have reproduced equation 5.6. In particular it is possible to appreciate as the potential difference changes with the passage of time. In this configuration* $\nu = 1 \times 10^8 Hz$.

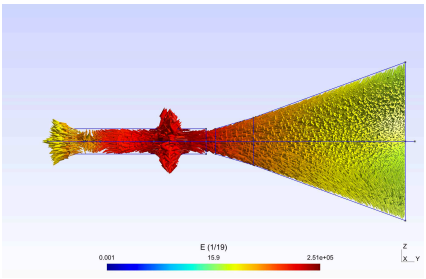
These rectangles, rotating around the y axis (that is the symmetry axis of the system), define the antenna, whose shape is that of a small cylinder. The potential difference varies between the peak values $[+1000V, -1000V]$ in the following way

$$V = -1000 \sin(2\pi\nu t + \phi) \tag{5.6}$$

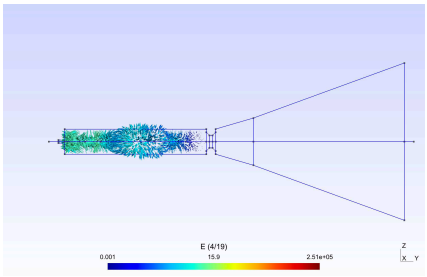
where ν is the external oscillation frequency, t is the simulation time and ϕ is the phase. In the following simulation ν has been blocked to

$$\nu = 1 \times 10^8 \text{ Hz} \longrightarrow T = \frac{1}{\nu} = 10^{-8} s \tag{5.7}$$

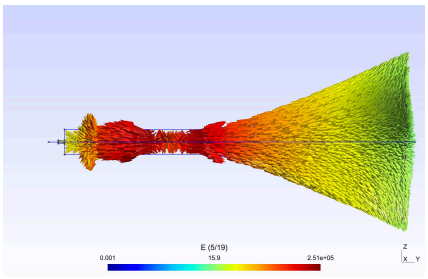
and the phase ϕ is set to 0.
In figure 5.5 we have reproduced equation 5.6, while in figure 5.6 it is possible to appreciate the effects of this oscillating potential difference on the electric field.
In particular, comparing figures 5.6a) and 5.6c), it is clear as the electric field changes direction due to the inversion of the potential difference. The inversion point, i.e. the time step in which the potential difference is equal to zero and therefore the electric field due to the external potential difference is null, corresponds to a simulation time of 5×10^{-9} s. See figure 5.5b).
The fact that the field is canceled at the fourth time step of figure 5.5, and not at the fifth time step, as expected from equation 5.6, is linked to the fact that the simulation starts with the step "zero" which corresponds to a simulation time equals to 1×10^{-9} s. For this reason the fourth "simulation step" corresponds to a simulation time of 5×10^{-9} s. These figures have been obtained with only few particles, in such a way we are able to appreciate only the electric field configuration due to the external potential difference, and no other effects due to the electric fields produced by plasma. The few loaded particles are responsible for the very small electric field in figure 5.6b); in fact, being the potential difference in this configuration equal to zero, the same electric field should be null.



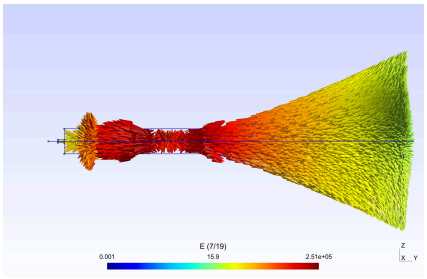
(a)Electric field at time-step number: 1



(b)Electric field at time-step number: 4



(c)Electric field at time-step number: 5



(d)Electric field at time-step number: 7

Fig. 5.6: *In these figures we have reproduced the different configurations of the electric field. In particular, comparing figures a) and c), it is clear as the electric field changes direction due to the inversion of the potential difference. In this case the frequency ν has been blocked to $\nu = 1 \times 10^8 Hz$. The inversion point, i.e. the time step in which the potential difference is equal to zero, corresponds to a simulation time of 5×10^{-9} s. See figure b). The fact that the field is canceled, in figure b), at the fourth time step, and not at the fifth time step, as expected from equation 5.6, is linked to the fact that the simulation starts with the step "zero" which corresponds to a simulation time equal to 1×10^{-9} s. For this reason the fourth "simulation step" corresponds to a simulation time of 5×10^{-9} s*

Let's start now with the analysis of the magnetic field.
The magnetic field is due to external permanent magnets (Samarium Cobalt Magnets) located near the dielectric tube in with flows the gas. Different configurations of magnetic fields have been reproduced in order to find the best experimental setup. In particular in figure 5.7 we have reproduced the last used configuration of the magnetic field. This figure can be seen as a section of the system of figure 5.4. In particular it is possible to appreciate the two magnets used with the magnetic field topology. In the plasma region is expected a magnetic field value that has an order of magnitude of $10^3 G = 0.1T$.
Figure 5.8 is a zoom of figure 5.7, but in this case we have also reproduced arrows to represent magnetic field direction.

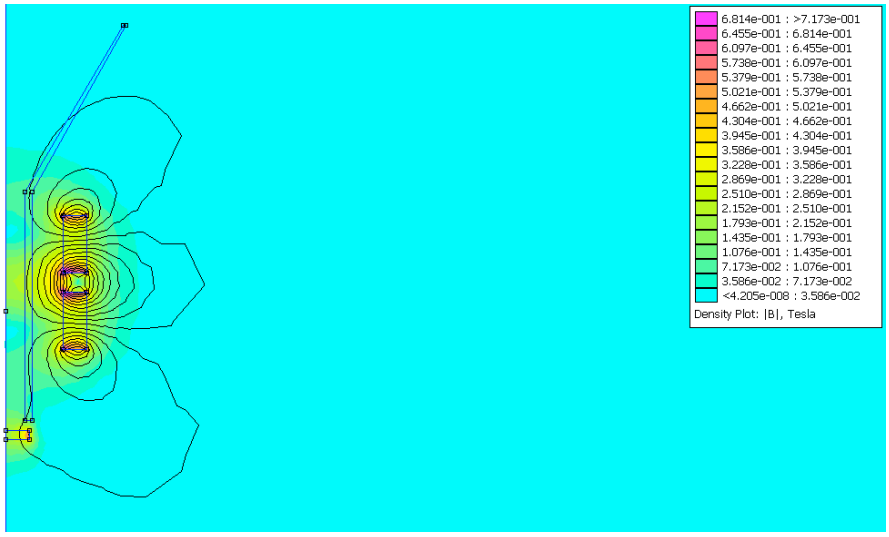


Fig. 5.7: *FEMM solver results for the magnetic field configuration reproduced in F3MPIC. In this figure it is possible to appreciate the spatial configuration of the external magnets. The problem has a cylindrical symmetry.*

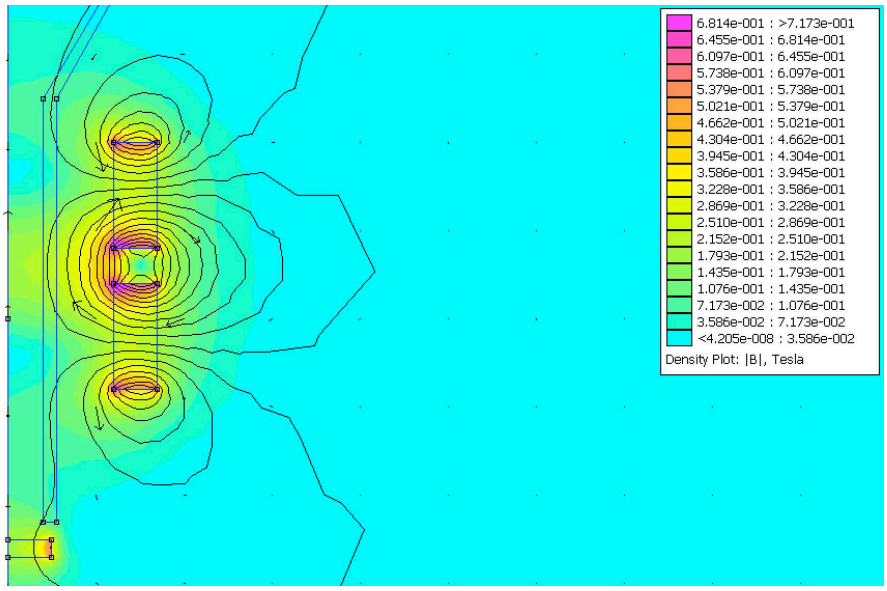


Fig. 5.8: *FEMM solver results for the magnetic field configuration reproduced in F3MPIC. Zoom of figure 5.7 with arrows representing magnetic field direction.*

In figures 5.7 and 5.8 we have used the results, obtained using an external solver called FEMM, to reproduced the magnetostatic field topology in F3MPIC. In figure 5.9 it is possible to appreciate the obtained results. In this case we have only reproduced a section of the studied system.

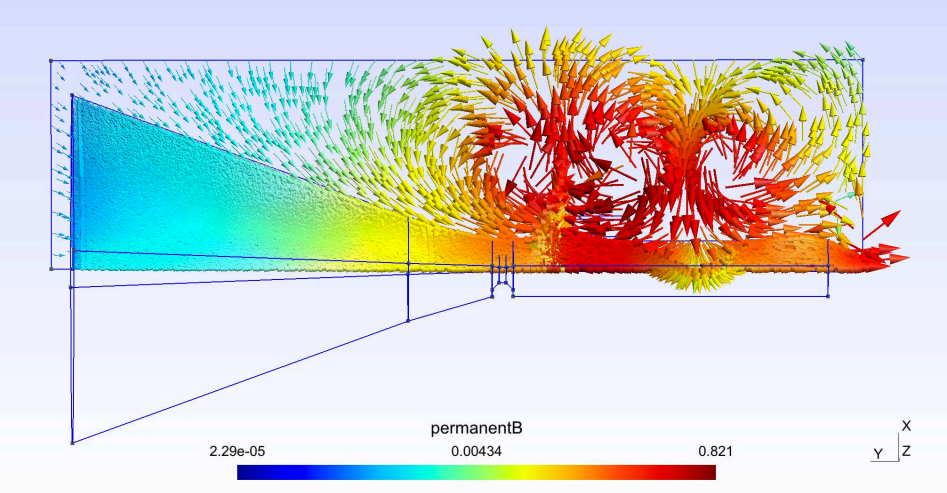


Fig. 5.9: *Section of the magnetic field configuration reproduced in F3MPIC using FEMM as external solver. The magnetic field is greater near the external magnets.*

5.4 Conclusions

In this chapter we have shown the obtained results for the electromagnetic fields used in the new HPT thruster prototype. To complete the characterization of the new thruster it should be necessary to perform some simulations working at the real experimental density, and to compute, with the new developed tools, some quantities like specific impulse and thrust. However, nowadays this characterization of the source is not feasible; in fact the real experimental plasma density is very high $\sim 10^{19} m^{-3}$ and it is difficult to work at this density keeping also sustainable computing times (In the current implementation F3MPIC runs only on a single processor).
In line of principle we could try to work at lower applied voltage and density and, with a scaling mechanism, we could try to make inferences on the expected values of some quantities at higher density. However it is not easy to find a pattern which, calculated at low density, it is also well verified at higher density.
For this reason we have decided to stop the simulation and to start the work to parallelize F3MPIC using MPI. In the months to follow we will complete this parallelization and the new code will be used to conclude the analysis started in this chapter.

6 A first integration of F3MPIC with ADAMANT: an electromagnetic solver

> *There is a single light of science, and to brighten it anywhere is to brighten it everywhere.*
>
> **— Isaac Asimov**

Abstract

In the previous chapters we have introduced different algorithms to improve the actual implementation of F3MPIC. In particular we have introduced a completely new MCC module, a new tracking method and a new charge conserving method. In the last chapter we have also applied these algorithms to an existing HPT source to continue its optimization phase. The main disadvantage of the actual implementation of F3MPIC is the solver used to manage electromagnetic fields; in fact nowadays F3MPIC uses an electrostatic solver called GETDP, an open source code. This one works only with charge density and not directly with current density to compute the new fields.
In order to obtain a computationally-efficient numerical tool to investigate the physics mechanisms related to electromagnetic wave propagation as well as a self-consistent plasma transport within the plasma source, it is necessary to side the old electrostatic solver with a new electromagnetic one called ADAMANT, a full wave numerical tool based on a set of coupled surface and volume integral equations developed at CISAS during the HPH.COM project. In this chapter we will develop some numerical tools to integrate the new electromagnetic solver in F3MPIC. The effective integration requires a more detailed study that will be addressed in the following years

6.1 Introduction

As already introduced in the previous chapter, recent advances in plasma-based propulsion systems have led to the development of electromagnetic (EM) Radio-Frequency (RF) plasma generation and acceleration systems called Helicon Plasma Thruster.

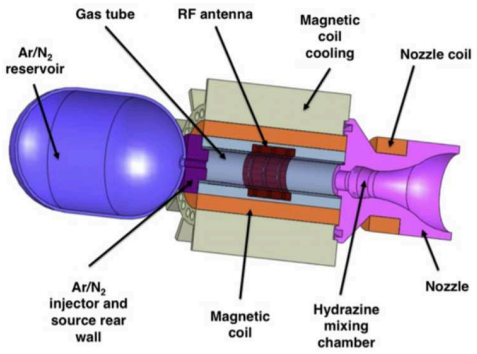


Fig. 6.1: *Artist's impression of the HPH.COM helicon plasma thruster.*

The main components of a helicon thruster, see figure 6.1, are: (i) a gas feeding system, (ii) an RF antenna, and (iii) magnetic coils. The feeding system injects a neutral gas into a dielectric cylindrical chamber, surrounded by a RF antenna system that works in the MHz range, ionizes the neutral gas and heats the resulting plasma. The magnetic coils provide the quasi-axial magnetic field that enables the propagation of helicon waves and the confinement of plasma inside the cylindrical source.
In order to optimize the propulsive figures of merit like thrust (i.e. plasma momentum) and specific impulse, in a HPT propulsion system it is convenient to distinguish two main stages, where different physical processes take place: the production stage in the plasma source, and the acceleration stage at the exhaust section of the thruster.
In the HPT electrical propulsion system of figure 6.1, the plasma source is a helicon plasma source, derived from industrial plasma sources. Its efficient plasma generation can realize high and variable specific impulses and good thrust efficiency, thus allowing the HPT to compete with other plasma thrusters. For these reasons, HPTs are subject of current research.
ADAMANT (Advanced coDe for Anisotropic Media and ANTennas) ([28]) was first developed for the analysis and design of radiofrequency antennas which drive the discharge in helicon plasma sources. It uses a set of coupled surface and volume integral equations in which the unknowns are the surface electric current density on the antenna conductors and the volume polarization current within the plasma. The latter can be inhomogeneous and anisotropic whereas the antenna can have arbitrary shape. The set of integral equations is solved numerically through the Method of Moments with sub-sectional surface and volume vector basis functions. This approach allows the accurate evaluation of the current distribution on the antenna and in the plasma as well as the antenna input impedance, a parameter crucial for the design of the feeding and matching network. For its high versatility ADAMANT was chosen as the best electromagnetic solver to be introduced in F3MPIC. The aim of the future research work will be to join these two codes, in particular in the next section we will explain in detail the integration model expected.

6.2 General scheme for the integration of the new electromagnetic solver in F3MPIC

In the previous section we have introduced ADAMANT, the aim of this section is to show the chosen scheme to bring up the integration between the existing time-domain electrostatic solver: GETDP, with the external frequency-domain electromagnetic solver: ADAMANT.
F3MPIC is a PIC code that is able to work with particles; in particular it is able to compute locally (i.e. on F3MPIC nodes) current and charge densities. It is also able to manage external magnetic and electric fields. However, as already explained, in its actual implementation F3MPIC isn't able to compute time varying components of electric and magnetic fields. These fields can be managed by ADAMANT. As clear these fields depend strongly from locally charge and current densities.
The main idea to obtain a fully electromagnetic PIC code is the following: use F3MPIC to compute the actual plasma parameters (e.g. plasma density, current density and electron temperature) and then give these values to ADAMANT, which solves the electromagnetic problem within the plasma discharge for the power deposited by the RF antenna into the plasma that, in turn, drives the plasma transport.
This approach will allow a self-consistent simulation of the electromagnetic fields and transport phenomena that occur within the discharge. In particular the electrostatic solver GETDP, already implemented in F3MPIC, will still be used to compute the electrostatic part of the electromagnetic fields and these fields will be added to their electromagnetic parts provided by ADAMANT. See figure 6.2 for clarity

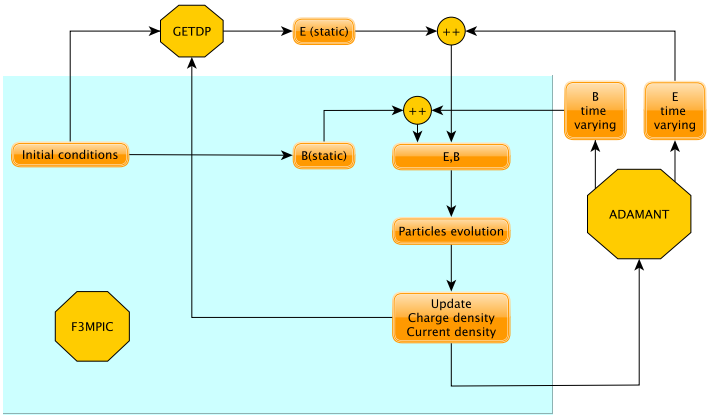


Fig. 6.2: *Scheme of F3MPIC-ADAMANT-GETDP integration*

The most important problems of this approach are the following

- ADAMANT in its actual implementation isn't able to accept this type of F3MPIC input.
- The mesh used in F3MPIC is not compatible with ADAMANT mesh. In fact ADAMANT works well only with a mesh with a nodes density much less dense than the one used by F3MPIC

The second problem in principle could be easily solved, in fact it is sufficient to work with two meshes with the same geometry but with different mesh fineness. To compute charge and current density on ADAMANT nodes starting from F3MPIC nodes, it is then sufficient to use an interpolation method as the one explained in the previous chapters.
The first problem, however, is not so easily solvable, and the integration of the nodal density computed on F3MPIC nodes, and then interpolated on ADAMANT nodes, will be part of a future research project; in fact to accept the new type of input it is necessary to modify heavily the current implementation of ADAMANT.
In the next section we will focus on the implementation of a new charge density scheme for the new electromagnetic solver in fact, as shown in figure 6.2, to compute the time varying electromagnetic fields it is necessary to compute not only current density but also charge density.

6.2.1 Implementation of a new charge density scheme for the new Electromagnetic Solver

In chapter 3 we have shown how it is possible to obtain an algorithm to achieve charge conservation. In particular we have seen how to implement a structured mesh and how to compute on the nodes of the structured mesh the vector $\mathbf{J}$.
To prepare F3MPIC for the future integration with the new electromagnetic solver ADAMANT, it is indispensable to compute on ADAMANT nodes current density vectors. To obtain this result, as anticipated in the previous section, it is only necessary to do a new IDW interpolation using, at this time, ADAMANT nodes as target elements i.e. elements in which it is necessary to compute current density, and not F3MPIC nodes as it has been done before. Remain only to compute charge density on ADAMANT nodes.
This problem is easier if compared with the previous one, in fact charge density is a scalar quantity. In the previous implementation of F3MPIC there was a subroutine that was used to compute charge density on F3MPIC nodes as shown in chapter 4. A first possibility could be to use this subroutine, in fact this one is fully compatible and it has several advantages like working directly with unstructured mesh.
In this section we want to introduce a new algorithm to calculate charge density. In this case the charge density is calculated firstly on the same structured mesh nodes used for current density deposition, and then, thanks to an interpolation, the charge density values are computed on ADAMANT nodes; in this way we don't use directly F3MPIC nodes.
Both these methods are consistent and they give essentially the same results. Generally each of these two methods have points of strength and weakness. The old one is generally preferred if you want to treat more consistently walls, conversely the new method is characterized by a high efficiency and a high speed of execution, in fact it is not necessary to calculate covolumes to compute charge density.
Let's show now the new algorithm used to compute charge density ρ on the structured cubic mesh. We will follow the same notation of chapter 3. The spatial profile of the charge density

$\rho(i,j,k)$ on node (i,j,k) is given by superposition of the charge of a particle assigned to each node[1] as

$$\rho(i,j,k) = \sum_{n=1}^{N_p} q_n p_{2c} S_i(x_n) S_j(y_n) S_k(z_n) \tag{6.1}$$

where the $S_i(*)$ function was defined in equation 3.12, N_p is the total number of the particles in the system, q_n is the charge assigned to particle n and p_{2c} is the number of computational particles assigned to each electron or ion as seen in chapter 2.
The procedure to find the "FALSE" node index from which to start the deposition of the charge densities is the same of chapter 3 and, as before, it is necessary to use firstly the FLOOR function. All the prescription of that chapter are always valid. Once the node (i,j,k) is found, the deposition starts, in particular in this configuration a particle produces a charge density variation on eight nodes.
For algorithm recursion we rewrite the $S_i(*)$ function using two new functions called W_i and W_{i+1} and defined as

$$W_i = i + 1 - \frac{x_n}{\Delta x} \tag{6.2}$$

$$W_{i+1} = \frac{x_n}{\Delta x} - i \tag{6.3}$$

In such a way the charge density can be computed as follows

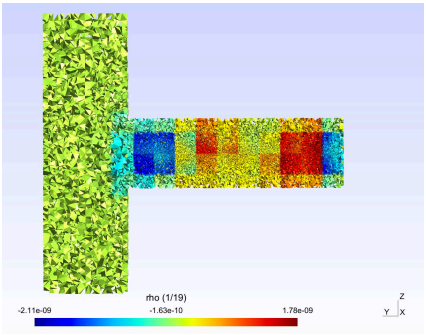
$$\begin{aligned}
\rho(i,j,k) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_i W_j W_k \\
\rho(i,j+1,k) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_i W_{j+1} W_k \\
\rho(i,j,k+1) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_i W_j W_{k+1} \\
\rho(i,j+1,k+1) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_i W_{j+1} W_{k+1} \\
\rho(i+1,j,k) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_{i+1} W_j W_k \\
\rho(i+1,j+1,k) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_{i+1} W_{j+1} W_k \\
\rho(i+1,j,k+1) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_{i+1} W_j W_{k+1} \\
\rho(i+1,j+1,k+1) &= \frac{1}{\Delta x \Delta y \Delta z} q_n p_{2c} W_{i+1} W_{j+1} W_{k+1}
\end{aligned} \tag{6.4}$$

As clear from equation 6.4, the charge density ρ is measured in $\frac{C}{m^3}$.
In the next figure we follow the evolution of the total charge density for twenty global time-steps, i.e we compute.

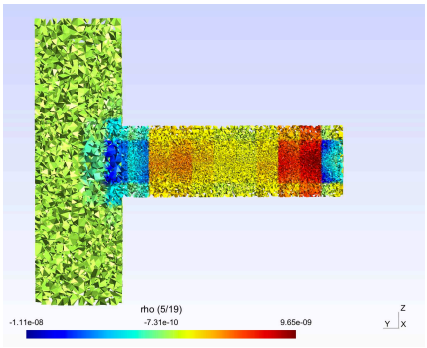
$$\rho_{\text{TOT}} = \rho_e + \rho_i = n_e q_e + n_i q_i \tag{6.5}$$

where n_e and n_i are the electron and ion densities. The studied system is the same of figure 3.7 already presented in chapter 3.

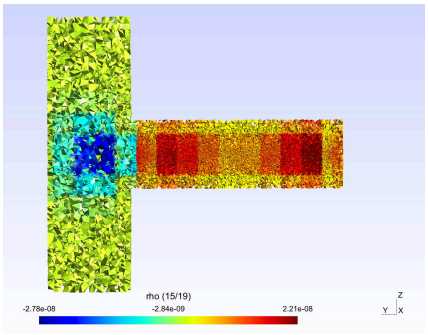
[1] For a detailed description of this equation see [38]. It can be proved that this equation in reality is equation 3.16. In particular this equation has been derived by Umeda following the same logical passages of section 3.3 applied, this time, to charge density instead of to the current density



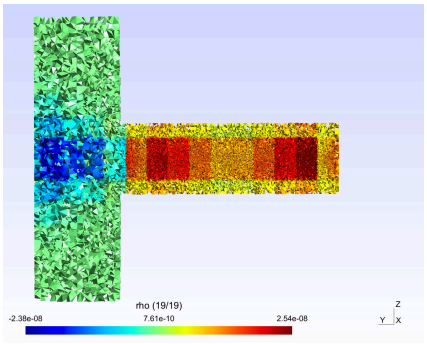
(a)Total charge density at time step number: 1



(b)Total charge density at time step number: 5



(c)Total charge density at time step number: 15



(d)Total charge density at time step number: 19

Fig. 6.3: *Evolution of charge density at different time-steps. In this case the charge density is firstly computed on the structured mesh points with the algorithm presented in equation 6.4, and then by interpolation is found in all F3MPIC nodes. The structured mesh used for the interpolation is enough visible in figure 6.3a) looking at the discontinuity in colors.*

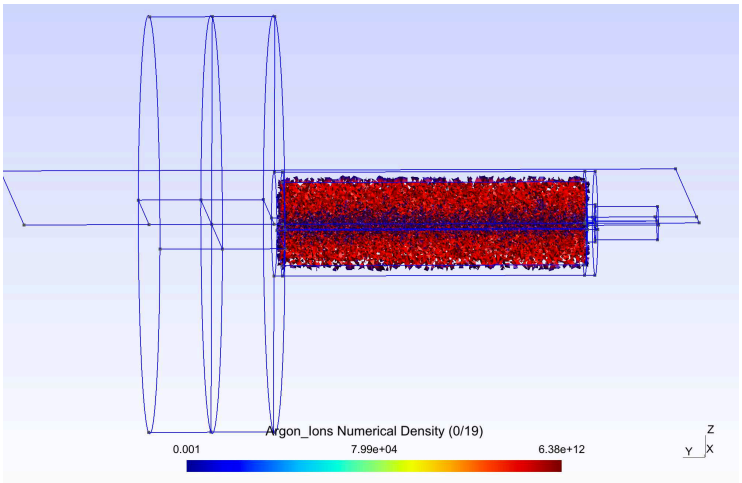


Fig. 6.4: *Ion density after 20 global time-steps computed using the charge deposition subroutine explained in chapter 4*

The values obtained in figure 6.3 with the new algorithm, are fully compatible with the expected ones; look, for example, at figure 6.4.
In this figure we have reproduced the ion numerical density after twenty global time steps

computed using the charge deposition subroutine explained in chapter 4. It is clear as the electrons leave system faster if compared with ions, due to their lower inerzia (after twenty global time-steps ions have not yet left the source). The peak of the numerical ion density in the source is

$$n_i \sim 10^{12} \frac{1}{m^3} \tag{6.6}$$

and it corresponds to an ion charge density of

$$\rho_i \sim 10^{-7} \frac{C}{m^3} \tag{6.7}$$

fully compatible with the peak value of the numerical ion density in figure 6.4d).
We conclude this section noting that in figure 6.3 it is also possible to appreciate the structured mesh in which the discontinuity in the colors shows the discretization with cubes. Remember that, using this algorithm, charge density was interpolated from structured mesh points to F3MPIC nodes.
In this particular case to reduce the number of computations, each cube of the structured mesh has a big edge. As showed in previous chapters a structured mesh that uses cubes with lower edges reduces interpolation errors and thus it improves algorithm efficiency.

6.3 Conclusions

In this chapter we have introduced the main idea that will be used to introduce in F3MPIC a new electromagnetic solver like ADAMANT.
In particular F3MPIC will be used to compute the actual plasma parameters (e.g. plasma density, current density and electron temperature). These values will be then given to ADAMANT, which solves the electromagnetic problem within the plasma discharge for the power deposited by the RF antenna into the plasma that, in turn, drives the plasma transport.
In this chapter we have proposed different solutions for some problems, that could arise during the effective integration. However the real integration of the two codes requires a more detailed study that will be addressed in the following years.
At the end of this chapter we have also proposed a new way to compute charge density using the same structured mesh defined in chapter 3.

7 Conclusions

> *Science cannot solve the ultimate mystery of nature. And that is because, in the last analysis, we ourselves are a part of the mystery that we are trying to solve.*
>
> **— Max Planck**

In the present study we have developed a new version of the 3D PIC code F3MPIC studying and validating new algorithms to manage the interactions between charged particles in such a way that a strong optimization of the existing STRONG hardware could be possible.
In particular in this thesis we have obtained the following results:

1. In the second chapter a completely new MonteCarlo code (MCC) has been introduced to simulate the interactions between charged particles and neutral ones. In particular a strong emphasis has be done on the new recombinational model, that is fully integrated with the new MCC code presented.

2. In the third chapter a new charge conservation method, fully compatible with standard PIC structure, has been deeply explained and tested. The new algorithm follows Umeda's paper and it is a revised version of it.

3. In the fourth chapter a new particle tracking algorithm, proposed recently by Haselbacher A. and others, has been implemented. In particular, in the new tracking algorithm, has also been proposed a revised version of the old F3MPIC charge deposition code, in which has been introduced a new subroutine to manage secondary electrons emission.

4. In the fifth chapter have been presented the results of some simulations to characterize the electromagnetic fields of the new HPT thruster. This thruster prototype is now in development at CISAS and its design is based on the scaling up of the prototype developed during the project HPH.COM.

5. In the sixth, and last chapter, a new numerical tool has been proposed to integrate the old electrostatic F3MPIC solver: GETDP, with the new electromagnetic one: ADAMANT. In particular a revised version of the algorithm to compute charge density on F3MPIC nodes has been deeply explained and tested.

If compared with previous F3MPIC algorithms or with others well established numerical tools, the new algorithms are generally faster and more efficient.
In the following years this new version of F3MPIC will be intensively used and, after completing the work needed to parallelize F3MPIC, a first strong experimental campaign

will be done with the aim to validate, also experimentally, the new code. In such a way the new developed tools will be at the basis of a plasma based research whose objective is to continue the study and the optimization of a customized high-power plasma source.
In particular this research activity will proceed through a methodology that will combine a theoretical investigation, a detailed numerical analysis, and an extensive experimental campaign to achieve the following high level objectives: (i) **physical investigation** into lagrangian charged particle interactions, (ii) development of others **numerical tools** and couple them with the just developed ones, (iii) **design, development, and test** of a high-power Helicon plasma source.
The objectives of each stage can be summarized as follows:

- **The physical investigation phase** will be used to find other methods and algorithms to improve the actual F3MPIC implementation. In particular the theoretical analysis, developed in chapter 3, will be used to study in detail the electromagnetic interactions among charged particles taking into account the presence of oscillating current sources like antenna or electromagnetic emitters.

- **The numerical analysis** will consist in: (ii) the effective parallelization of F3MPIC using MPI, (ii) the effective implementation of a method to integrate the existing time-domain electrostatic solver with an external frequency-domain electromagnetic one as proposed in chapter 6, (iii) the implementation of the convergence iterative interaction between the two solvers.

- **The experimental setup** will result in high-reconfigurable, high-efficient and high-power plasma source addressed to space propulsion systems. Especially the experimental analysis will be conducted by a deep scan of different parameters, e.g., magneto-static field, plasma density, RF input power, antenna geometry, and it will be featured several diagnostic systems. A Langmuir probe will be used to characterize the plasma discharge in terms of: plasma density, electron temperature, and ion species, while the specific impulse and thrust efficiency of the thruster will be measured by means of a Faraday probe. All the measured experimental quantities will be compared with the simulated ones obtained, partially, in chapter 5 by means of the new algorithms developed in chapters 4 and 6.
  Testing will be done at the experimental facility of CISAS-University of Padua.

As clear from the previous itemize, in all the proposed stages the new version of F3MPIC, developed in this thesis, will be central and it will guide the experimental campaign.
In particular a deep numerical analysis will be carried out, not only for verification and validation of the code, but also for testing the prediction.
Further optimization of the performance of the source will be also evaluated in relation to the propulsive figure of merit (e.g. thrust efficiency, specific impulse) obtained installing the source in a HPT and different application of the source will be studied. After the tests the final project will lead to obtain a good experimental configuration not only for a HPT thruster, but also for the improvement of other type of thrusters like for example an ion or Hall thruster.

Collaborations

This work has been performed in collaboration with T4I S.r.l., which is a spin-off of the University of Padua under the supervision of Dr. Marco Manente.

Bibliography

[1] Safdar Ali. *Electron - ion recombination data for plasma applications*. Department of Physics, Stockholm University, 2012 (cit. on p. 20).

[2] D. Banas, P. Jagodzin, M. Pajek, A. Gumberidze, and A. Surzhykov. „Monte-Carlo simulations of the radiative recombination of ions with electrons in cold magnetized plasma“. In: *The Royal Swedish Academy of Sciences* (2014) (cit. on p. 19).

[3] E. M. Baroody. „A Theory of Secondary Electron Emission from Metals“. In: *Phys. Rev.* 78 (1950) (cit. on p. 68).

[4] William T. Vetterlin Brian P. Flannery Saul A. Teukolsky. *Numerical Recipes in C: The Art of Scientific Computing*. William H. Press, 1992 (cit. on p. 46).

[5] Jaume Navarro Cavallè, Eduardo Ahedo, Mario Merino, Victor Gomez, and Mercedes Ruiz. „Helicon Plasma Thrusters: prototypes and advances on modeling“. In: *IEPC-285*. 2013 (cit. on pp. 83, 84).

[6] Nicolas Chenouard and Others. „Objective comparison of particle tracking methods“. In: *Nature methods* 11.2 (2014) (cit. on p. 55).

[7] R. Chordà, J.A. Blasco, and N. Fueyo. „An efficient particle-locating algorithm for application in arbitrary 2D and 3D grids“. In: *International Journal of Multiphase Flow* 28 (2002) (cit. on p. 56).

[8] M. S. Chung and T. E. Everhart. „Simple calculation of energy distribution of low energy secondary electrons emitted from metals under electron bombardment“. In: *J. Appl. Phys.* 45 (1974), p. 707 (cit. on p. 70).

[9] Birdsall C.K. and Langdon A.B. *Plasma Physics via Computer Simulation*. Adam Hilger Bristol, Philadelphia, and New York, 1991 (cit. on pp. 8, 15).

[10] Stéphane Cordier, Thierry Goudon, Michaël Gutnic, and Eric Sonnendrücker. *Numerical Methods for Hyperbolic and Kinetic Problems*. Vol. 7. European Mathematical Society, 2005 (cit. on pp. 36–40).

[11] Shepard Donald, ed. *A two-dimensional interpolation function for irregularly-spaced data*. Vol. pp. 517–524. Proceedings of the 1968 ACM National Conference (cit. on p. 46).

[12] P. Dular and C. Geuzaine. *GetDP reference manual: the documentation for GetDP, a general environment for the treatment of discrete problems*. `http://getdp.info` (cit. on p. 3).

[13] Chen Francis F. *Introduction to Plasma Physics and Controlled Fusion*. 1 vols. Plenum Press, 1974 (cit. on p. 15).

[14]A. L. Fabris, C. V. Young, M. Manente, D. Pavarin, and M. A. Cappelli. „Ion Velocimetry Measurements and Particle-In-Cell Simulation of a Cylindrical Cusped Plasma Accelerator“. In: *IEEE Transactions on Plasma Science* 45.1 (2015) (cit. on pp. v, 3).

[15]Andrea Lucca Fabris. „Experimental Characterization of Plasma Sources for Space Propulsion“. PhD thesis. SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE TECNOLOGIE E MISURE SPAZIALI INDIRIZZO DI MISURE MECCANICHE PER L'INGEGNERIA E LO SPAZIO (cit. on p. 4).

[16]Jurgen Geiser and Sven Blankenburg. „Monte Carlo Simulations of Elastic Scattering with Applications to DC and High Power Pulsed Magnetron Sputtering for TI3SIC2“. In: *ComputerPhysics Communications* (2011) (cit. on pp. 7, 9, 13, 14).

[17]Christophe Geuzaine and Jean Francois Remacle. „Gmsh A 3D finite element mesh generator with built in pre and post processing facilities“. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009). Ed. by Ltd. John Wiley et Sons (cit. on p. 3).

[18]A. Haselbacher, F.M. Najjar, and J.P. Ferry. „An efficien and robust particle-localization algorithm for unstructured grids“. In: *Journal of computational physics* 225 (2007), pp. 2198–2213 (cit. on pp. 5, 55–58, 62, 66).

[19]Umran S. Inan and Marek Golkowskj. *Principles of Plasma Physics for Engineers and Scientist*. Cambridge University Press, 2011 (cit. on p. 15).

[20]T. Ito and M. A. Cappelli. „Electrostatic probe disruption of drift waves in magnetized microdischarges“. In: *Applied Physics Letters* 94 (2009) (cit. on pp. 25, 32).

[21]T. Ito and M. A. Cappelli. „High Speed Images of Drift Waves and Turbulence in Magnetized Microplasmas“. In: *IEEE Transactions on Plasma Science* 1228 - 1229 (36) (cit. on pp. 25, 32).

[22]Villasenor J. and Buneman o. „Rigorous charge conservation for local electromagnetic field solvers“. In: *Comput. Phys. Commun.* 69 (1992), pp. 306–316 (cit. on p. 35).

[23]S. K. Karkari, H. Kabariya, S. Jain, et al. *Observation of counter propagating E x B drifts in the magnetized plasma column of a dc magnetron source*. Tech. rep. Institute for Plasma Research Bhat Gandhinagar Gujarat India (cit. on p. 49).

[24]H. A. Kramers. „On the theory of X-ray absorption and of the continu- ous X-ray spectrum“. In: *Philos. Mag., 46:836* (1923) (cit. on p. 20).

[25]W. Kraus. „RF Ion Sources“. Max Planck Institut fur Plasmaphysik (cit. on p. 85).

[26]MICHAEL A. LIEBERMAN and ALLAN J. LICHTENBERG. *Principles of Plasma Discharges and Materials Processing*. Wiley, 2005 (cit. on p. 14).

[27]M. Manzolaro, M. Manente, D. Curreli, et al. „Off-line ionization tests using the surface and the plasma ion sources of the SPES project“. In: *Rev. Sci. Instrum.* 83, 02A907 (2012) (cit. on p. 3).

[28]D. Melazzi and V. Lancellotti. „ADAMANT: A surface and volume integral-equation solver for the analysis and design of helicon plasma sources“. In: *Computer Physics Communications* (2014) (cit. on p. 94).

[29]Eastwood J. W. The virtual particle electromagnetic particle-mesh method. „The virtual particle electromagnetic particle-mesh method“. In: *Comput. Phys. Commun.* 64 (1991), pp. 252–266 (cit. on p. 43).

[30]D. Pavarin, F. Ferri, M. Manente, et al. „Development of plasma codes for the design of mini-helicon thrusters 32th International Electric Propulsion Conference“. In: *IEPC*. 2011 (cit. on pp. v, 3).

[31]M. C. Pinto, S.Jund, S. Salmon, and E. Sonnendrucker. „Charge conserving FE-PIC codes on general grids“. In: *hal-00311429v2* (August 17, 2008) (cit. on pp. 34, 35).

[32]Chen X. Q. „An efficient particle-tracking algorithm for two-phase flows in geometries using curvilinear coordinates“. In: *Numerical Heat Transfer* 31 (1997), pp. 387–405 (cit. on p. 56).

[33]Courant R., Friedrichs K., and Lewy H. „On the Partial Difference Equations of Mathematical Physics.“ In: *BM J. 11, 215-234* (1967) (cit. on p. 22).

[34]S. A. Schwarz. „Application of a semi empirical sputtering model to secondary electron emission“. In: *J. Appl. Phys.* 68 (1990), p. 2382 (cit. on p. 68).

[35]H. Seiler. „Secondary electron emission in the scanning electron microscope“. In: *Journal of Applied Physics* 54 (1983) (cit. on pp. 68, 71).

[36]Francesco Taccogna, Savino Longo, and Mario Capitelli. „Plasma-surface interaction model with secondary electron emission effects“. In: *Physics of Plasmas* 11 (2004), p. 1220 (cit. on pp. 68, 69).

[37]T. Umeda, Y. Omura, T. Tominaga, and H. Matsumoto. „A new charge conservation method in electromagnetic particle-in-cell simulations“. In: *ComputerPhysics Communications* (2003) (cit. on pp. 34, 35, 43, 44).

[38]Takayuki Umeda. „Study on Nonlinear Processes of Electron Beam Instabilities via Computer Simulations“. PhD thesis. Department of Communications and Computer Engineering Graduate School of Informatics Kyoto University, Kyoto, JAPAN, 2003 (cit. on pp. 5, 34, 43, 44, 97).

[39]V. Vahedi and M. Surendra. „A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges“. In: *ComputerPhysics Communications* (1994) (cit. on pp. 5, 7, 8, 13).

[40]Hockney R. W. and Eastwood J.W. *Computer Simulation Using Particles*. Adam Hilger Bristol, Philadelphia, and New York, 1988 (cit. on pp. 15, 33).

[41]D. W. Rule Y. Hahn. „Direct radiative capture of high-energy elec- trons by atomic ions“. In: *J. Phys. B. At. Mol. Phys., B10:2689* (1977) (cit. on p. 20).

[42]R. H. Pratt. Y. S. Kim. „Direct radiative recombination of electrons with atomic ions: Cross sections and rate coefficients.“ In: *Phys. Rev. A,27:2913* (1983) (cit. on p. 20).

List of Figures

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses a modified version of the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.