

Loughborough University
Wolfson School of Mechanical, Electrical
and Manufacturing Engineering

Advance Electron Beam Sources

Prateep Ganesh Kasinathan

December 2021

Abstract

Text of the Abstract.

Contents

Abstract	i
1 Introduction	1
1.1 Motivation	1
1.2 Novelty	2
1.3 Goal	2
2 Background Theory	4
2.1 Plasma Discharge	4
2.1.1 Paschen's Law	5
2.1.2 DC Discharge	6
2.1.3 AC Discharge	9
2.2 Electron Beam Guns	10
2.2.1 DC-based Devices	10
2.2.2 AC-based Devices	13

3	Particle-in-Cell Simulations	14
3.1	Particle Mover	16
3.2	Charge and Current Deposition	19
3.3	Field Solver	21
3.4	Field Interpolation	23
3.5	Monte-Carlo Collisions	23
4	XOOPIC	26
4.1	Overview	26
4.1.1	Remote Server	27
4.1.2	Input Files	28
4.2	Improvements	29
4.2.1	Motivation	29
4.2.2	Methodology	30
4.2.3	Validation	33
4.2.4	Future Work	36
5	Starter Simulations	37
5.1	Split ring resonator	38
5.2	Simulations	39
5.2.1	Parameters	40
5.2.2	Results	41

List of Tables

List of Figures

2.1	Paschen curve for Helium, Neon, Argon, Hydrogen, and Nitrogen gases [1].	6
2.2	Circuit diagram with a source voltage (V_s) and variable resistor (R) to control the current through a discharge region (C to A) [1].	7
2.3	Schematic highlighting the regions present in a DC glow discharge [1]. The cathode is on the left, the grounded anode is on the right. (CDS is the cathode dark space, NG is the negative glow, and AZ is the anode dark space.	8
2.4	Comparison between a parallel plate (left) and a hollow cathode (right).	10
2.5	Electron Gun Design by Kornilov et al [1]: (1) hollow cathode, (2) anode, (3) permanent magnet, (4) auxiliary electrode (emidng cathode), (5) emission channel, and (6) accelera8ng electrode.	11
2.6	Electron Gun Design by Bakeev et al [1]: (1) electron beam, (2) enclosure to maintain forevacuum-pressure, (3) hollow cathode, (4) anode, (5) emission plasma, (6) metal disk, (7) emission channel, (8) extractor, and (9) focusing magnetic system. (Note: the un-labelled components were for testing the beam.)	12
2.7	Electron Gun Design by Del Pozo et al [1]: a plasma chamber (1) gas feed inlet, (2) gas valve, (3 and 4) discharge electrodes, and (7) accelerating electrodes. . .	13
3.1	Process flow of PIC simulations [1].	15
3.2	Visualisation of rotation from Boris algorithm [1].	18

3.3	Visualisation of bilinear interpolation []	19
3.4	Illustration of ADI method [].	22
3.5	Addition of null collision to produce a constant collision frequency across all energies [].	24
5.1	Schematic (left) and photo (right) of SRR [].	38
5.2	Cross section of SRR used for XOOPIC simulations.	39
5.3	Time series plot of ions across different discharge gap widths.	41
5.4	Time series plot of electrons across different discharge gap widths.	41

Chapter 1

Introduction

Additive manufacturing (AM), better known as 3D printing, is not a new technology, being established in the 1980s. Early AM processes only used plastics as the printing material, with metal 3D printing being later established in the mid-1990s. Over the years, new AM processes for metal have been developed; however this report will focus on a particular process called Electron-beam additive manufacturing (EBAM).

1.1 Motivation

EBAM utilises a layer-by-layer approach to produce the final print. As the name alludes, an electron beam gun is used to melt a layer of metal powder on a bed. Once solidified, more powder feedstock is added and the process repeats to produce the subsequent layers. To generate the electron beam, a filament (generally tungsten) is heated to a point where it begins to emit electrons via thermionic emission. Once emitted, these electrons are then focused by a magnetic field into a single beam.

Using a filament-cathode electron gun however comes with two major shortcomings. The first issue is with relation to the short life time of the filaments themselves, which is problematic for larger scale printing with much longer print times. The second is to do with the slow thermal

response times when heating and cooling the filaments. This isn't necessarily an issue for the generation of continuous electron beams. Nevertheless with pulsed electron beams, the lengthy response time limits the pulse duration hence the maximum energy that can be delivered.

1.2 Novelty

The most conventional solution to combat the short filament life times is to use an alternative filament material that have a lower evaporation rate. Lanthanum hexaboride (LaB6) and Cerium hexaboride (CeB6) are two such alternative very commonly used in electron microscopy. While these materials do increase the life time, the problem of lengthy thermal response durations still remains.

The solution proposed in this project is to forego the filament-cathode model in favour of a plasma-cathode electron gun. This model alleviates the issue of filament life span since the electron gun can continue operation so long as there is a steady source of plasma. Plasma-cathode electron guns also have a much faster thermal response since it is not relying on the process of thermionic emission for electron generation.

1.3 Goal

The goal of this project is to first perform simulations on potential electron beam gun designs using kinetic description plasma models; specifically using particle-in-cell simulations. This allows multiple parameters of the electron beam gun device to be tested, reducing the need physical prototypes and saving costs. Once a potential device has been successfully simulated, a scaled prototype will be produced and evaluated. The findings will be compared to the simulation results to ensure the device functions as intended.

The rest of this report is broken as follows. A introduction of plasma and plasma-cathode electron guns can be found in chapter 2. Chapter 3 goes on to provide an overview of particle-in-cell simulations. Then, chapter 4 provides a brief description of the simulation program

being used, XOOPIC. Finally, the results of some starter simulations can be found in chapter 5.

Chapter 2

Background Theory

Plasma is often described as the fourth state of matter. It has similar characteristics to gases, where it is compressible and does not conform to a specific shape; however the key difference is that plasmas are highly electrically conductive, even capable of producing it's own magnetic field. This is because plasmas contain a large number of positive ions that interact in a “sea” of disassociated free-moving electrons.

2.1 Plasma Discharge

Plasma are generated in one of two primary ways. The first is via extreme heating of a gas whereby the electrons gain sufficient energy to escape the electromagnetic force (EMF) of the nucleus. The most obvious example of this is in stars where the interstellar gas clouds collapse and form a plasma for nuclear fusion. The other method, which is the main focus of this report, is the exposure of a gas to a large voltage difference, which in turn which in turn ignites the plasma. An everyday example of this is lightning, where a charge build-up between the clouds-to-ground grows until the air in between breaks down.

2.1.1 Paschen's Law

The voltage necessary to breakdown a gas is given by Paschen's law. It states that breakdown voltage is a function of two parameters: the pressure of the gas and the distance between the electrodes, referred to as the gap length. Specifically, the breakdown voltage is given by the product of these two parameters.

In order for a breakdown to occur, there needs to be a small number of electrons already present in the gas. This can be caused internally by a smaller number of already excited gas molecules, or externally by light entering the gas chamber disassociating a few electrons. Then by applying a voltage, these electrons gain energy creating other electrons from ionising collisions. When more electrons are generated from the collisions than are lost, an avalanche is created, which in turn causes a breakdown of the gas.

The voltage breakdown can expressed by the following equation:

$$V_B = \frac{B\rho d}{\ln(A\rho d) - \ln[\ln(1 - \frac{1}{\gamma_{se}})]} \quad (2.1)$$

where V_B is the breakdown voltage, ρ is the pressure of the gas, d is the gap length, γ_{se} is a coefficient for secondary-electron emissions (explained in section 2.1.2), and A and B are constants for a given that are determined experimentally.

However it much easier to understand Paschen's law pictorially. Figure 2.1 illustrates the voltage breakdown curves across various gases. Each curve is slightly different, however all of them do exhibit the shape of a parabola. Supposing if:

- **The gap length remains constant.** Starting with a large pressure, the mean free path of an electron within the gas is low, meaning it does not have sufficient time to gain enough energy to cause an ionising collisions with neutral gas. As the pressure is then reduced, the mean free path increases, making it easier for the electrons to generate ionising collisions; until a certain critical pressure that is. Beyond this, decreasing the pressure further decreases the likelihood of an electron colliding with a neutral gas particle.

- **The pressure remains constant.** When the gap length is very small, the electrons do not have sufficient time to accelerate the energy required to ionise the background gas. Increasing the gap to a certain point gives the electrons sufficient time to gain the energy required for ionisation. However, as the gap length continues to be increased, the electric field between the electrodes decreases, hence the electrons do not have enough energy to cause ionising collisions.

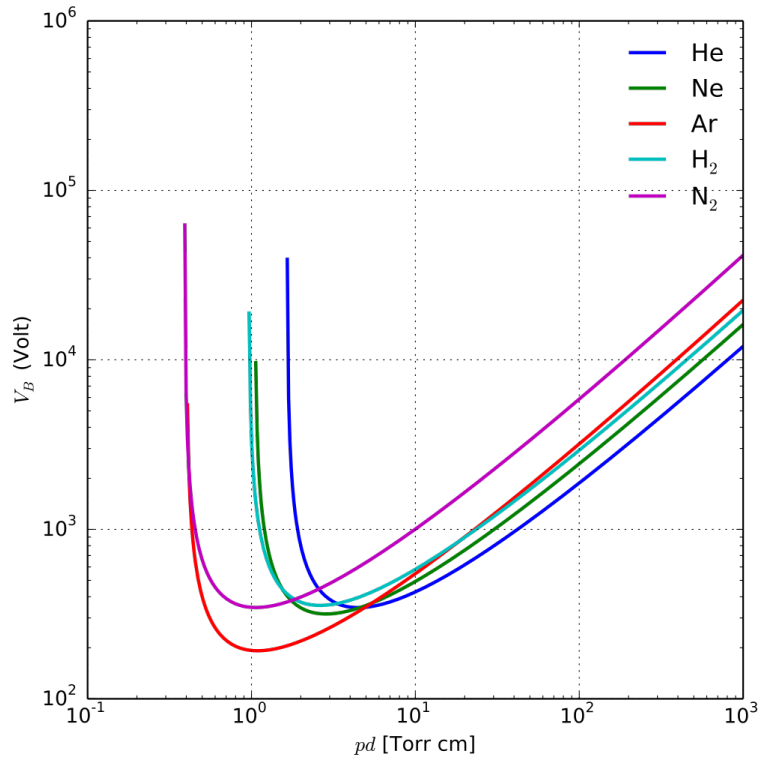


Figure 2.1: Paschen curve for Helium, Neon, Argon, Hydrogen, and Nitrogen gases [1].

One thing to note about Paschen's law is that it loses its accuracy for smaller gap lengths, below 10 to 15 μm . The model incorrectly predicts that the voltage breakdown rises towards infinity; however in reality, there is an initial increase before it decreases again.

2.1.2 DC Discharge

Consider a circuit as seen in figure 2.2. Two parallel electrodes with a DC voltage applied, and a neutral gas contained within a chamber. As the resistance of the variable resistor is

decreased, which in turn increases the current through the plasma, one would observe three distinct discharge regions.

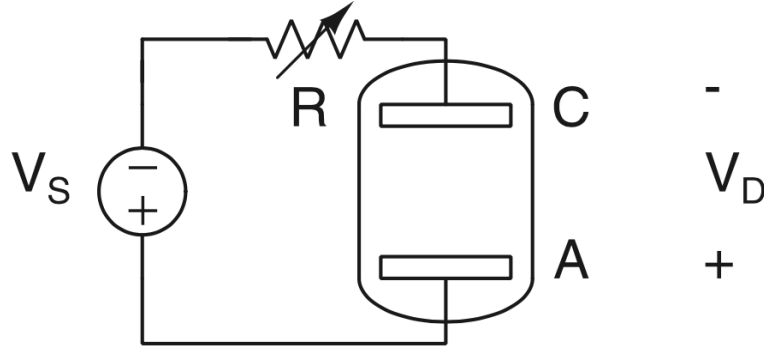


Figure 2.2: Circuit diagram with a source voltage (V_s) and variable resistor (R) to control the current through a discharge region (C to A) [].

The first, is the dark discharge (or sometime referred to as the Townsend discharge) region. Initial, the voltage between the electrodes builds up as the only current through the plasma is caused by pre-existing electrons, say from cosmic radiation; however this current quickly saturates. Then, once the electrons gain sufficient energy, they can begin colliding with the background gas to produce additional electrons in a process called the Townsend avalanche. Once this avalanche can be self-sustaining, the voltage breakdown of the gas is achieved.

After the breakdown, the plasma is said to be in a glow discharge region. Here, the voltage across the plasma decreases as a transition from a gas to plasma state causes a decrease in resistance implying that ionisation process from the avalanche is more efficient. This efficiency stems from the fact that the electrons are generated by another means in addition to the standard Townsend avalanche. This process is called the secondary-emission of electrons, which is caused the energetic collisions of ions or metastable with the surface of the cathode. These secondary-electrons are then accelerated by the electric field and cause further Townsend avalanches.

Initially, ion bombardment on the surface of the cathode is non-uniform but as the current generated from this increases, it eventually stabilises and distribution of the plasma (and thus the ions) across the cathode become more uniform. This is referred to as subnormal glow and normal glow respectively. As the current is increased further, ion bombardment across the cathode becomes saturated as it covers the entire surface of the cathode. This is referred to as

abnormal glow, and increasing the current further causes the glow discharge to become an arc.

In the arc discharge region, the ion bombardments onto the cathode cause the cathode to heat up to a point where electrons are generated via thermionic radiation. This significantly reduces the resistance of the plasma, causing a very large drop of the voltage.

For the purposes of this project, all plasma will be operating in the glow discharge region. The reason being, operating beyond the point abnormal glow increases the sputtering rate. Sputtering is the ejections of ions from the cathode caused by the ion bombardment process. While useful for processes such as ion etching, sputtering would not be favourable for the application of a plasma cathode electron gun as the consumption of the cathode is something to be eliminated. Sputtering has the additional downside of potentially contaminating the final print as ejected atoms could end up in the electron beam.

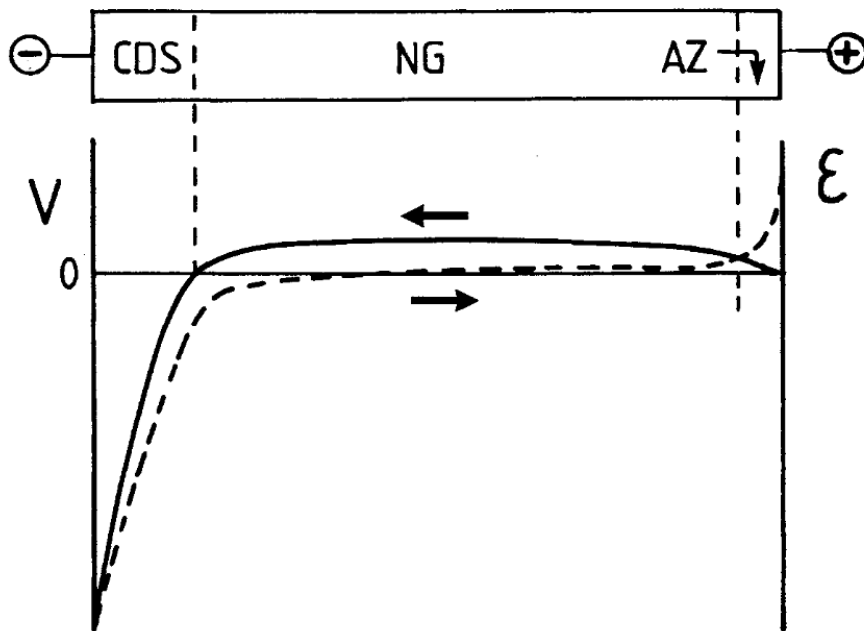


Figure 2.3: Schematic highlighting the regions present in a DC glow discharge [1]. The cathode is on the left, the grounded anode is on the right. (CDS is the cathode dark space, NG is the negative glow, and AZ is the anode dark space).

In a glow discharge there are typical three spatial regions present. These include a cathode dark space, a negative glow region, and the anode dark space. This can be observed in figure 2.3, that shows the potential difference in each region. Do note, as the distance between the electrodes is increased, additional regions may develop, however these three regions will always

persist. The dark space regions are called *sheaths* while the negative glow region is known as the *bulk plasma*. Generally, the sheaths on the cathode will be much larger than that of the anode, as it corresponds to the region where electrons are being accelerated before gaining sufficient energy to cause ionising collisions. In contrast, the anode sheaths are caused by electrons colliding with the anode, thus creating a region of positive charge which repel any ions to the bulk plasma. Finally, the bulk plasma is the quasi-neutral region that contains the ions and electrons of the plasma.

2.1.3 AC Discharge

If the voltage source in the circuit of figure 2.2 were to be replaced with a low frequency AC source, the discharge behaviour would be effectively identical to that of the DC discharge. This is provided that the half time period of an AC cycle is larger than the duration for ions and electrons to move across the electrodes.

However, as the frequency of the AC source is increased, typically to the region of radio or microwave frequencies, there is an asymmetry between the movement of the ions and electrons. The electrons are capable of responding to the change in the electric fields relatively quickly; however, due to the ions being significantly heavier than electrons, they have a much slower response time hence are restricted to their inertia.

Since the ions cannot respond to the changing electric field quick enough, the generation of electrons via the process of secondary-emissions is minimal. Instead what occurs is that electrons begin accelerating through the bulk plasma towards the anode during the first half period of the AC signal. Then as the direction of the electric field reverses in the second half period of the signal, the positions of the anode and cathode flip, and any electron that has not collided with the original anode (which is now the cathode), gets accelerated through the bulk plasma towards the new anode. This oscillating behaviour increases the likelihood of ionising collisions with the neutral background gas. Hence it is no surprise that increasing the frequency of the AC source decreases the breakdown voltage of the plasma.

2.2 Electron Beam Guns

2.2.1 DC-based Devices

As seen in the previous section, a parallel plate configuration is one possible geometry to produce a electrons via a DC discharge. Another commonly used alternative is the hollow cathode geometry. Hollow cathodes are typically cylindrical in nature, where the anode remains the same but the cathode has been replaced with cup-like shape that is hollow in the centre (hence the name). Nonetheless, other designs are possible with examples of square hollow cathode developed by Fukuda et al [1] and also tapered cathode shapes by Ohtsu et al [2]. In truth, the exact hollow cathode geometry to be used highly depends on the intended use case as each design has its pros and cons.

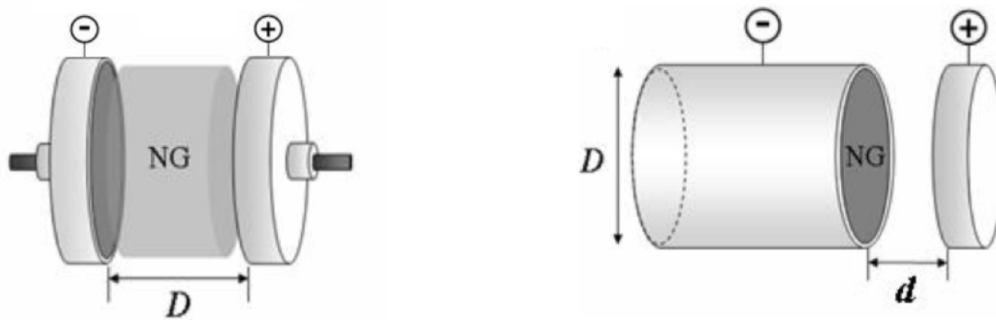


Figure 2.4: Comparison between a parallel plate (left) and a hollow cathode (right).

Despite the designs used, all hollow cathodes take advantage of a phenomenon known as the hollow cathode effect. There are multiple factors that contribute to the hollow cathode effect. However, it is generally agreed upon that the primary mechanism is caused by the pendulum effect of electrons. Electrons, generated via secondary-emissions from the cathode, tend to oscillate back and forth between the cathode walls. This motions of electrons increases the likelihood that any given one will undergo an ionising collision with a neutral gas atom [3]. This is quite similar to the behaviour of of AC discharge mentioned in section 2.1.3, however the difference being it only oscillates between the single electrode. The downside of this behaviour is that there tends to be excess heating of the cathode and increased sputtering potential form ion bombardments on the cathode. However, these drawbacks can be minimised by carefully

selecting design and operating conditions of the hollow cathode. The benefits over a parallel plate geometry are that hollow cathodes tend to have a lower breakdown voltage (particularly at low gas pressures) [] compared to their parallel plate counterparts, and generally produce a higher current density for a given operating voltage []. As such, it is the hollow cathode design that is the preferred geometry for a plasma-cathode electron gun.

An example of a hollow cathode based electron gun can be in figure 2.5. This device was developed by Kornilov et al [], and can be divided into two separate sections: the discharge region and acceleration region.

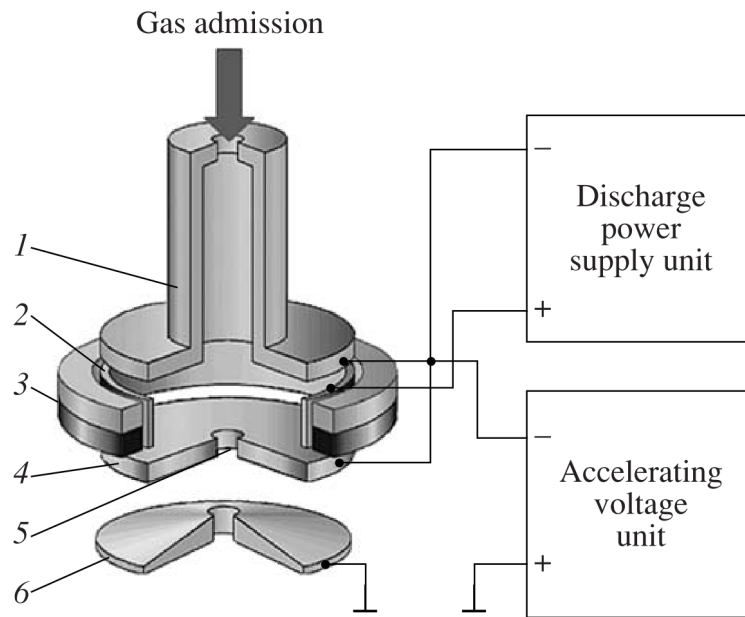


Figure 2.5: Electron Gun Design by Kornilov et al []: (1) hollow cathode, (2) anode, (3) permanent magnet, (4) auxiliary electrode (emidng cathode), (5) emission channel, and (6) accelera8ng electrode.

The electrodes (1 and 2) used in the discharge region. There are two key differences seen here compared to the hollow cathode shown previously. First and foremost is that the anode takes the shape of a ring rather than a plate, which is to simply to allow the electrons generated to exit the discharge region. The other difference is the presence of an inlet at the top of the cathode. This allows a constant stream of gas feed to replenish the gas that has been ionised. This gas feed also ensures that pressure within the discharge region remains roughly constant, at about 1-10 Pa. There is also a permanent magnet (3) used to focus the beam into the emission channels (5). As for the accelerating region, there are two addition electrodes (4

and 6) with a high voltage applied to accelerate beam out of the gun. The pressure in the acceleration is kept quite low, at around 0.01 Pa. This mitigates electric charges within the gun and also prevents the focused beam from diverging.

The device by Kornilov et al. produced an electron beam with a diameter roughly 260 μ m. This was achieved with a power of 3kW to the overall gun. The potential difference for the discharge electrodes were approximately 350-450 V, whilst the voltage at the accelerating electrodes were on the order of 30 kV.

Some other approaches for designing an electron gun exists, such as the one developed by Bakeev et al [1] seen in figure 2.6. This design forgoes the accelerating region, instead relying on much higher voltages for the discharge electrodes. This was between 20-30 kV. The pressure of the discharge region was also higher, at around 10-30 Pa.

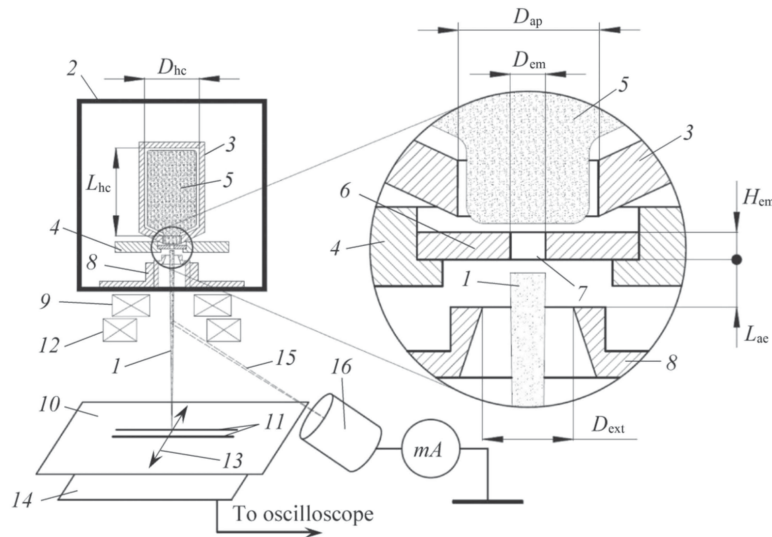


Figure 2.6: Electron Gun Design by Bakeev et al [1]: (1) electron beam, (2) enclosure to maintain forevacuum-pressure, (3) hollow cathode, (4) anode, (5) emission plasma, (6) metal disk, (7) emission channel, (8) extractor, and (9) focusing magnetic system. (Note: the un-labelled components were for testing the beam.)

One of the biggest differences in this design was that the hollow cathode was that the aperture of the hollow cathode was reduce to a gap of 8 cm. This was done to improve the electron extraction efficiency, and as a result the hollow cathode was less cup-shaped and more akin to a water bottle. Another difference is that magnets used for focussing the electron beam was located outside the gun itself. This could possibly because the hollow cathode design used

already focused the electron beam out of the gun, hence the magnets are only necessary to obtain a smaller beam diameter. The final diameter of the electron beam was 200 μm .

2.2.2 AC-based Devices

Because an AC-based plasma does not rely on the secondary emissions from the cathode, the utilisation of a hollow cathode design is not strictly necessary. Despite that, the general structure of the AC electron guns do share a number of similarity to their DC-based counterparts.

An example of a radio frequency (RF) plasma cathode electron gun can be seen in figure 2.7, developed by Del Pozo et al [1]. In the figure 2.7, there are a number of parts that pertain to the RF circuit, however the components within the plasma device (seen in the dashed outline 12) are quite familiar.

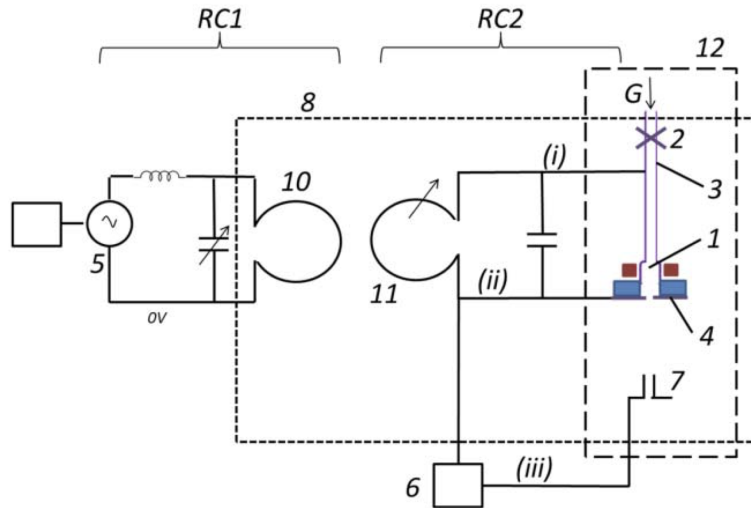


Figure 2.7: Electron Gun Design by Del Pozo et al [1]: a plasma chamber (1) gas feed inlet, (2) gas valve, (3 and 4) discharge electrodes, and (7) accelerating electrodes.

For the RF plasma device, there is a gas feed inlet (2) that maintains a relatively high pressure within the plasma chamber (1), between 10-100 Pa. There are also two sets of electrodes. The first set are the discharge electrodes (3 and 4) designed for frequency of 84 MHz. The generated electrons then exit the plasma chamber into the accelerating electrodes. These electrodes have a very large voltage of -60 kV, and the overall region of was kept at a low pressure of 0.01 Pa in order to ensure the electrodes were electrically isolated.

Chapter 3

Particle-in-Cell Simulations

Plasmas are complex non-linear systems and contain many unknown variables. Therefore, simplified models are often used to capture major physics of the system whilst ignoring details that can be considered negligible. There are several types of plasma models but the most commonly used methods are fluid description models and kinetic description models [1].

The fluid description method aims to generalise the plasma quantities, such as its density and velocity, by averaging them over velocity space within 2D or 3D region [1]. This is achieved by numerically solving the fluid equations; obtained by utilising the velocity components of the Boltzmann equation [1]. Then, the electromagnetic fields are obtained by combining the fluid equations with Maxwell's equation.

In contrast, kinetic description models track the position and velocities of particles within the plasma, taking into account the electromagnetic forces acting on them. As a general rule of thumb, fluid models simulate the plasma behaviour over a macroscopic scale; whilst kinetic models highlight plasma behaviour at the microscopic level.

Particle-in-cell (PIC) simulations are a variant of kinetic description models that tend to be favoured because of its easy formulation. Real-world plasma systems contain a prodigious number of particles, which include electrons, ions, and the background neutral gas; thus, tracking all these particles would be immensely computationally taxing. PIC simulations partially solve

this by tracking so-called *super-particles*. Each super-particle is scaled to represent a number of “real” particles. This scaling factor does not affect the trajectory of the super-particles as the trajectory of particles within a electromagnetic field are only governed by its mass and charge (see the Newton-Lorentz expression in equation 3.3). For the rest of this chapter, the term super-particle and particles are used interchangeably.

For PIC simulations, the particle properties (i.e. their position and velocity) are defined in continuous space within the simulation domain. However, the electromagnetic field values are only specified at fixed grid points. Therefore, intermediary steps are required to discretise charges and current densities onto the grid and also interpolation of forces by fields on the particles. These particles and fields are updated sequentially, with time being advanced in small discrete constant steps. An overview of this procedure can be seen in figure 3.1. The rest of this chapter is dedicated to providing an overview each process in the flow chart.

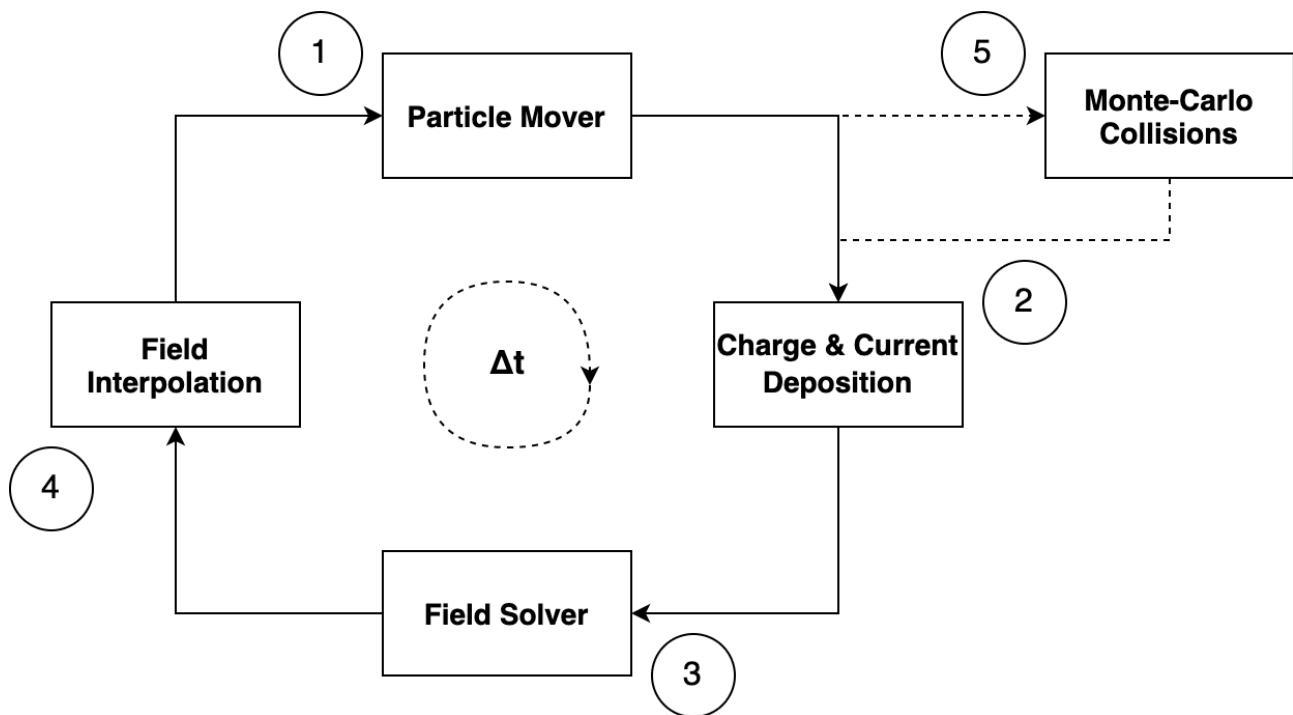


Figure 3.1: Process flow of PIC simulations [1].

3.1 Particle Mover

When first launching the simulation, a random distribution of particle positions and velocities is initialised within the simulation domain. The acceleration of a particle is given by Newton's second law:

$$\frac{d\mathbf{v}}{dt} = \frac{\mathbf{F}}{m} \quad (3.1)$$

The force acting on the particle is determined by the Lorentz force:

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (3.2)$$

where \mathbf{E} and \mathbf{B} are the electric and magnetic fields respectively, and q corresponds to the charge of the particle.

Combining equations 3.1 and 3.2 gives the Newton-Lorentz equation:

$$\frac{d\mathbf{v}}{dt} = \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (3.3)$$

As for the particles velocity, it is given by:

$$\frac{d\mathbf{s}}{dt} = \mathbf{v} \quad (3.4)$$

In order to obtain the particle velocity and position, equations 3.3 and 3.4 have to be numerically integrated with respect to time using the *finite-difference method*; specifically the *leapfrog method*. Thus, the new finite-difference forms of the respective equations are:

$$\frac{\mathbf{v}^{t+\frac{1}{2}} - \mathbf{v}^{t-\frac{1}{2}}}{dt} = \frac{q}{m}(\mathbf{E} + (\frac{\mathbf{v}^{t+\frac{1}{2}} + \mathbf{v}^{t-\frac{1}{2}}}{2}) \times \mathbf{B}) \quad (3.5)$$

$$\frac{\mathbf{s}^{t+1} - \mathbf{s}^t}{dt} = \mathbf{v}^{t+\frac{1}{2}} \quad (3.6)$$

Notice that the particle position is given by the *forward-difference form*, whereas its velocity is given by the *central-difference form*. This is done for the symplectic nature of the leapfrog method, conserving the energy of the dynamic system. An intuitive explanation for this is that when solving for the updated position of the particle, its average velocity should be used. Using either the initial or final velocities can result in the particle gaining energy if its accelerating, or losing energy if its decelerating.

To solve the Newton-Lorentz equation, the de-facto method used is the *Boris algorithm* [] for its accuracy, speed, and simplicity. The method first redefines both velocity terms as:

$$\mathbf{v}^{t-\frac{1}{2}} = \mathbf{v}^- - \frac{q\mathbf{E}}{m} \frac{dt}{2} \quad (3.7)$$

$$\mathbf{v}^{t+\frac{1}{2}} = \mathbf{v}^+ + \frac{q\mathbf{E}}{m} \frac{dt}{2} \quad (3.8)$$

When substituted back into the the finite-difference Newton-Lorentz equation (equation 3.5, this eliminates the electric field, resulting in an equation that expresses the rotation of the particle's velocity:

$$\frac{\mathbf{v}^+ - \mathbf{v}^-}{\Delta t} = \frac{q}{2m} (\mathbf{v}^+ + \mathbf{v}^-) \times \mathbf{B} \quad (3.9)$$

Because it describes a rotation, it can be solved geometrically. Using the following 2D case seen in figure x., it can be determined that the angle through which the velocity rotates is:

$$\tan\left(\frac{\theta}{2}\right) = \frac{\mathbf{v}^+ - \mathbf{v}^-}{\mathbf{v}^+ + \mathbf{v}^-} = \frac{q\mathbf{B}}{m} \frac{\Delta t}{2} \quad (3.10)$$

This angle can also be represented in vector form, referred to as \mathbf{T} . Using this vector form, a bisecting vector \mathbf{v}' can be obtained is parallel to the $\mathbf{v}^+ + \mathbf{v}^-$ vector (the sum of the pre-rotation and post-rotation vector), and perpendicular to both the \mathbf{T} vector (the scaled magnetic field) and the $\mathbf{v}^+ - \mathbf{v}^-$ vector (the difference between the post-rotational and pre-rotational velocity).

$$\mathbf{v}' = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{T} \quad (3.11)$$

Finally to obtain the $\mathbf{v}^+ - \mathbf{v}^-$ vector, this intermediate vector \mathbf{v}' is combined with a new vector \mathbf{S} which can be thought of as a scaled version of vector \mathbf{T} seen in equation 3.13.

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}' \times \mathbf{S} \quad (3.12)$$

$$\mathbf{S} = \frac{2\mathbf{T}}{1 + \mathbf{T}^2} \quad (3.13)$$

Thus, in order to determine the new velocity of the particle:

1. Calculate the velocity of the particle over the first half time step using equation 3.7.
2. Perform the full rotation of the velocity due to the magnetic field using equations 3.11 and 3.12.
3. Calculate the velocity of the particle over the second half time step using equation 3.8.

To obtain the particles new position, substitute the newly calculated velocity of the particle into equation 3.6. Once calculated, the positions of each particle in the simulation can be updated.

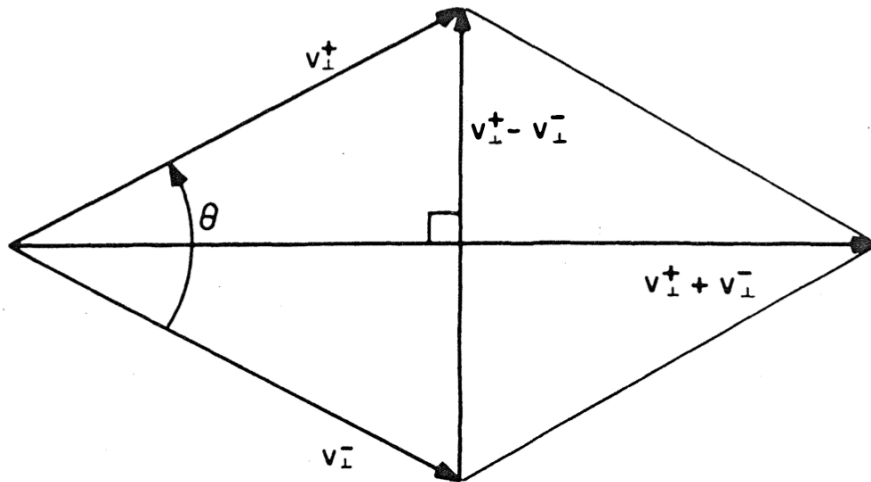


Figure 3.2: Visualisation of rotation from Boris algorithm [1].

3.2 Charge and Current Deposition

Once the particles have been moved, their charge and current densities are discretised onto a the specified grid. The exact method of discretising these properties depends on the type of grid for the PIC simulation. The one described in this report is for a uniform 2D rectilinear grid, known as the *bilinear interpolation method*.

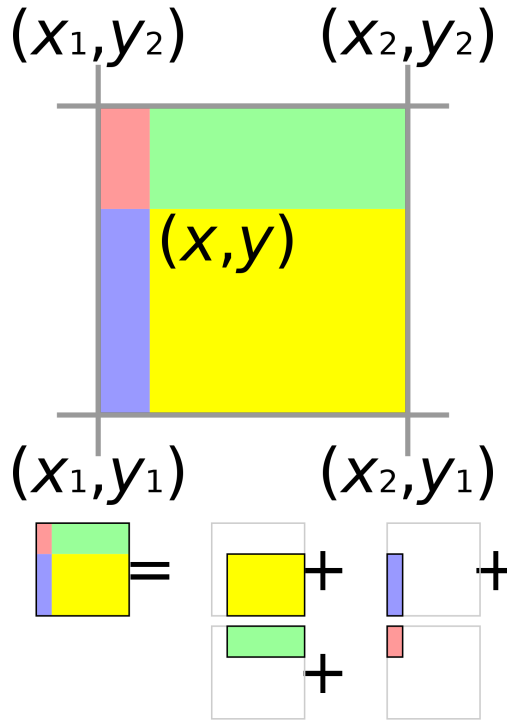


Figure 3.3: Visualisation of bilinear interpolation [1].

Consider the following case in figure 3.3. The charge of the particle within the grid cell (hence the name *particle in cell*) is distributed across the four nodes of the grid. The total charge experienced by each node must equal to the charge of the particle, with nodes closest to the particle experiencing the greatest weight. Thus the charge of the particle can be rewritten as a weighted mean as follows:

$$q = w_{11} \cdot q_{11} + w_{12} \cdot q_{12} + w_{21} \cdot q_{21} + w_{22} \cdot q_{22} \quad (3.14)$$

When solving this linear system, the resulting weights simply to:

$$\begin{aligned}
 w_{11} &= \frac{(x_2 - x) \cdot (y_2 - y)}{dx \cdot dy} \\
 w_{12} &= \frac{(x_2 - x) \cdot (y - y_1)}{dx \cdot dy} \\
 w_{21} &= \frac{(x - x_1) \cdot (y_2 - y)}{dx \cdot dy} \\
 w_{22} &= \frac{(x - x_1) \cdot (y - y_1)}{dx \cdot dy}
 \end{aligned} \tag{3.15}$$

Intuitively, the weight of a node corresponds to the ratio of the area of the rectangle opposite (formed by the point of the particle and the point of the opposite node) and the total area of the cell. Once the (weighted) charges at each node point has been calculated for all particles, the charge density of the node can be determined by:

$$\sum^n \frac{q_{i,j}^n}{dx \cdot dy} \tag{3.16}$$

where $q_{i,j}^n$ is the charge of a given particle n at the grid point i, j .

The area of the cell around the node is $dx \cdot dy$, with the exception being nodes at the edges or corners of the simulation domain. With such cases, the resulting area is halved and quartered respectively.

A similar process is done to determine the current densities around a node:

$$\sum^n \frac{q_{i,j}^n \mathbf{v}^n}{dx \cdot dy} \tag{3.17}$$

where the additional \mathbf{v}^n term is its average velocity vector of a given particle.

3.3 Field Solver

The field solver utilises the charge and current densities at the grid points to determine the electric and magnetic field values at corresponding grid points by solving Maxwell's equations. This report will only focus on the process of computing the electric field values.

Gauss's law dictates that the divergence of the electric field in a region is proportional to its charge density:

$$\nabla E = \frac{\rho}{\varepsilon_0} \quad (3.18)$$

The electric field is determined as the gradient of the electric potential, and by substituting that into the above equation results in Poisson's equation:

$$-\nabla^2 \phi = \frac{\rho}{\varepsilon_0} \quad (3.19)$$

Considering a 2D case with a cartesian coordinate system, Poisson's equation can be rewritten as:

$$\frac{\delta^2 \phi}{\delta x^2} + \frac{\delta^2 \phi}{\delta y^2} = -\frac{\rho}{\varepsilon_0} \quad (3.20)$$

In order to solve Poisson's equation in a discretised domain, the finite-difference method is employed. Specifically, using the central-difference, the resulting approximation seen in equation 3.21 is produced that is second order accurate.

$$\frac{\phi_{x+\Delta x} - 2\phi_x + \phi_{x-\Delta x}}{\Delta x^2} + \frac{\phi_{x+\Delta y} - 2\phi_y + \phi_{y-\Delta y}}{\Delta y^2} = -\frac{\rho}{\varepsilon_0} \quad (3.21)$$

There are several method to solve such an elliptical equation numerically. An overview of method used by the simulation program in this paper (see chapter 4) is described below.

The poisson equation is considered a form of an elliptic partial differential equation, which can be simplified in this form:

It is possible to rewrite this as a diffusion equation as seen in equation x, where as $t \rightarrow \infty$, then $\frac{du}{dt} \rightarrow 0$.

Applying the finite difference method to the equation above results in the following:

From this, a technique known as the *alternating-direction implicit* (ADI) method can be employed. In essence, the ADI utilises a concept known as operator splitting, wherein the du/dx and du/dy terms can be decoupled, then each can be solved by iterating over half a time step. So for the first half time step, a pass is performed along the x direction; then at the second half time step, a pass is performed along the y direction as illustrated in figure 3.4.

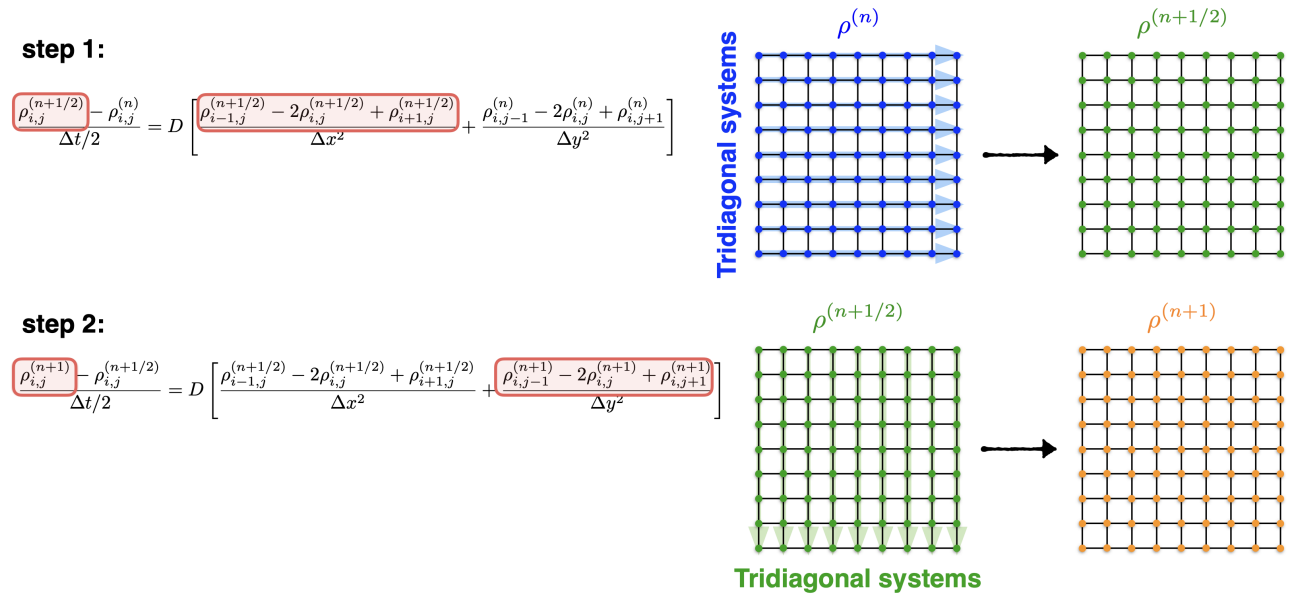


Figure 3.4: Illustration of ADI method [1].

The benefit of this method is that rather than solving a large sparse matrix of a 2D system, a set of two independent 1D systems can be solved. These 1D systems can be arranged as a tridiagonal matrix, which can be efficiently solved using the *Thomas algorithm* [1].

Once the poisson equation is solved and the electric potentials at the grid determined, the electric field can be calculated again by the finite difference method. As the electric field is a vector, it needs to be calculated both along the direction of the x-axis and y-axis. Considering

only the direction along the x-axis, equation 3.22 can be used. For the case of a boundary, equation 3.23 should be used instead.

$$E_x^{i,j} = \frac{\phi_{i+\Delta x,j} - \phi_{i-\Delta x,j}}{2\Delta x} \quad (3.22)$$

$$E_x^{i,j} = \frac{\phi_{i+\Delta x,j} - \phi_{i,j}}{\Delta x} \quad (3.23)$$

3.4 Field Interpolation

The interpolation of the the electromagnetic field values on the the particles is also performed using bilinear interpolation, however the reverse method to that done for the charge and current deposition. Therefore the the electric field experienced by the particle is given by the contribution of the electric field vectors at each grid point. This can be expressed as:

$$\mathbf{E}^n = w_{11} \cdot \mathbf{E}_{11} + w_{12} \cdot \mathbf{E}_{12} + w_{21} \cdot \mathbf{E}_{21} + w_{22} \cdot \mathbf{E}_{22} \quad (3.24)$$

where \mathbf{E}^n is the electric field vector on particle b .

For the magnetic field, the exact same set of equations are used, with the only difference being the electric fields at the grid points replaced with the corresponding magnetic field values. This process is repeated for all particles and the new electromagnetic field values are used in the particle mover for the next time step.

3.5 Monte-Carlo Collisions

Steps 1-4 are used to simulate collisionless plasma systems, whereby the coupling between particles are insignificant due to the large number of particles within their Debye sphere. In

essence, the particle interactions are primarily governed by the electromagnetic forces. However, when the elastic and inelastic collisions between particles can't be ignored, an additional step is sandwiched between steps 1 and 2 to account for this.

This is known as the *Monte-Carlo collision* (MCC) method. This describes the collisions between the particles probabilistically. To start, the particles are evaluated for collisions. If a collision occurs, the particle velocity is updated in lieu with the type of collision. One potential downside to using the MCC method is that source particles collide with a target “cloud”, which is not a simulated particle. This means, unless special care is taken to update the target, energy of the system is not conserved. Fortunately, the simulation program used for this report (seen in chapter 4) does take into account the.

Determining if each and every particle in the simulation undergoes a collision can be very computationally expensive, as this requires the computation of each particles kinetic energy and their corresponding collision cross sections. Therefore, a computational trick is employed by adding an artificial collision term who's cross section is chosen to force the total collision frequency for each species to be uniform and independent of the kinetic energy of the particle. This is called the *null collision*, which is illustrated in figure 3.5.

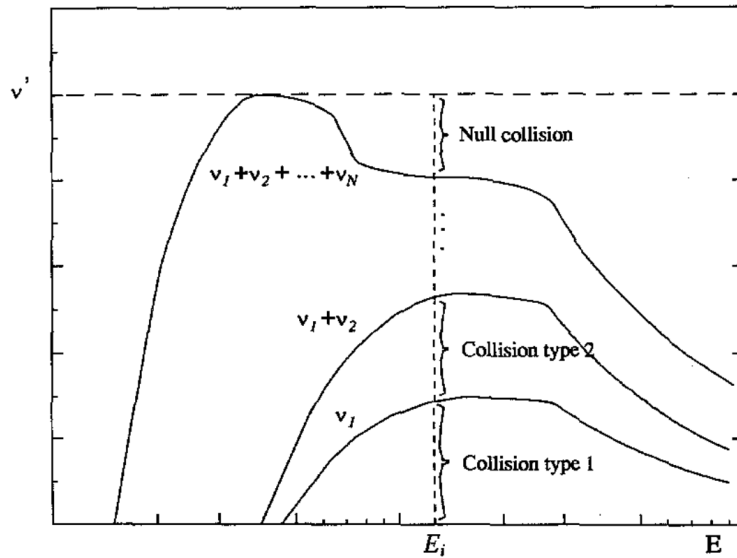


Figure 3.5: Addition of null collision to produce a constant collision frequency across all energies [1].

From this, the portion of particles that should undergo a collision can be determined by the

following expression:

$$P(t) = 1 - e^{-\nu t} \quad (3.25)$$

where ν is the maximum of the sum of all collision frequencies. This can be expressed mathematically in equation 3.26. Here, n_g is the neutral gas density, which can be assumed to be constant, σ_{total} is the total cross section from all collisions modelled (not including the null collision), and v is the relative speed between the particles and the target.

$$\nu = n_g \cdot \max(\sigma_{total}, v) \quad (3.26)$$

Once the number of particles that experience collisions have been determined, particles are randomly picked from the population of simulated particles. This way, the kinetic energy of only a small subset of particles from the pool will be determined, making the simulation more efficient. Once a particle has been chosen and its kinetic energy calculated, a random number between $[0, 1]$, called R , is selected. This is used to determine the collision type for the particle, as a function of the value of R , as shown by table 3.5.

$$\begin{array}{ll} R \leq \nu_1(\mathcal{E}_i)/\nu' & \text{(Collision type 1)} \\ \nu_1(\mathcal{E}_i)/\nu' < R \leq (\nu_1(\mathcal{E}_i) + \nu_2(\mathcal{E}_i))/\nu' & \text{(Collision type 2)} \\ \vdots & \\ \sum_{j=1}^N \nu_j(\mathcal{E}_i)/\nu' < R & \text{(Null collision)} \end{array}$$

If the collision type selected is a null collision, then it is treated as no collision occurred.

Chapter 4

XOOPIC

The PIC simulation software used in this project is called *X11 object-oriented particle-in-cell* (XOOPIC). It is a C/C++ based program that simulates 2D plasma systems. It was originally developed in the 1990s by the Plasma Theory and Simulation Group from the University of California at Berkley. The code base was subsequently forked by Plasma and Pulsed Power Group at Loughborough University, and improved upon to include additional output diagnostics for analysis.

4.1 Overview

XOOPIC is primarily a command-line-interface (CLI) program but support X11 windowing to visualise certain diagnostics in real time. Some examples of these real time diagnostics include: average kinetic energies of all the particles, phase-space plots of the position and velocities of the various particle species, and the magnitude and direction of electric field (at the grid points) within the simulated system. Whilst useful for determining the behaviour of the simulated plasma system, running the simulation with these windows does significantly reduce its performance. Therefore, XOOPIC's in built visualisations are only used to observe the snapshot behaviour of the simulated system, whilst any diagnostic information that is required for a further analysis is typically written onto output files.

Despite not running with X11 windowing, running these simulations continuously with XOOPIC typically takes a long period of time, on the order of days to weeks, before the simulation stabilises. This is because the simulations need to be run at a very small time step in order to be stable (see the next section for more information). As such, all XOOPIC simulations are run on remotely on Loughborough University's high performance servers. Multiple simulations are run concurrently in order to test various model parameters.

4.1.1 Remote Server

As mentioned above, XOOPIC's simulations are run remotely via secure shell (SSH). There are several risks associated with running the simulations remotely. The first being that if (and when) the connection between the client and the server is severed, the simulation will be killed by the server. This problem is alleviated with the use of terminal multiplexors, such as *GNU Screen*. This allows for the creation of multiple pseudo-terminal sessions that run in the background. Once created, the client is free to attach and detach sessions without the need for an uninterrupted connection to the server.

A broken client-server connection is not the only way a simulation can be terminated. Some others include unexpected server restarts, power outages, or overuse of server resources. In such cases, the simulation can be resumed using XOOPIC's *dump files*. These files can be thought as periodic saves of the simulations and can be used to restart them if the need arises. The interval between creating new dump files is set when starting the simulation. Though tempting to set this interval as small as possible in order to mitigate any data loss, generating these files is very expensive computationally as it involves writing most of the simulation variables onto disk. Thus, the dump file interval are set in the neighbourhood of hundreds of thousand time steps.

4.1.2 Input Files

The parameters for XOOPIC's simulations are set using an input file. This is simply a custom formatted text file, with a pseudo-JSON-like structure. The input file is divided into three sections: headers, variables, and region. The first two sections are for the user's benefit, describing the simulation and the parameter values used respectively.

The region section however is where the true simulation parameters are specified. It can be further divided into multiple subsections. The list of all possible subsections is vast, though the ones required by every simulation include:

- **Grid** where the dimensions of the plasma device are specified, either in cylindrical or cartesian geometry.
- **Control** where the control parameters such are set, such as the size of the time step or a flag to use a specific field solver.
- **Species** which state properties of the particle species being simulated.
- **MCC** that determine the collision characteristics of the plasma based on the background gas pressure and temperature. Collisions can also be turned off outright.
- **Load** that establishes the region in which to load the particle species.
- **Diagnostic** which as the name suggests extracts the various diagnostic information and saves them to a specific file.

The other subsections not listed are the various boundary parameters. The usage of these parameters will vary based on the type of device being simulated. Some of examples of included boundary parameters within XOOPIC include: a grounded boundary, a current source, and a dielectric boundary.

When constructing the input files, care must be taken to ensure that the grid size (within the grid subsection) and the time step (within the control subsection) obey a set of conditions so that the simulation is stable.

The grid size of the simulation, Δx and Δy should adhere to the equation:

$$\Delta x < 3.4\lambda_D \quad (4.1)$$

where λ_D is the Debye length.

As for the time step, Δt , it should fulfil the equation:

$$\Delta t < 0.2\omega_{pe}^{-1} \quad (4.2)$$

where ω_{pe} is the plasma frequency. Note, this condition is only valid for electrostatic plasma simulations.

4.2 Improvements

During the course of this project, the several improvements were made to XOOPIC. This primarily included bug fixes, but also the inclusion of a new boundary parameter (a subsection within the region section) called the *Circuit* boundary.

4.2.1 Motivation

XOOPIC previously had two types of (traditional) input source boundaries. An ideal voltage source, known as the *Equipotential* boundary; and an ideal current source, called the *Current Source* boundary.

With the Equipotential boundary, the simulation always keeps the potential at the boundary equal to the source voltage specified in the input file. This has the issue that there are no maximum or minimum bounds for current through the boundary. Thus, theoretically the current through the boundary grow exponentially to infinity. In practice however, as ions or electrons

collide with the boundary, a large but finite current is achieved. Nonetheless, it is possible for this current to exceed the rated current of the intended power supply to be used.

Likewise with the Current Source boundary, the simulation attempts to keep the current across the boundary at a constant specified value despite the potential at the boundary. And in practice, it is possible that the potential produced by the simulation exceeds the voltage that can be delivered by the power supply.

Hence, in order to accurately simulate a “real world” power supply, the Circuit boundary was introduced. This boundary represents the configuration of an ideal voltage source with a resistor attached in series. An illustration of this can be seen in figure x. The one significant assumption with this boundary is that the circuit shares the same ground plane as the rest of plasma device being simulated.

By adding the series resistor, in effect the maximum voltage and current across the boundary has been limited. As seen in figure x, the maximum voltage is obtained when the simulated device is an open circuit; and the maximum current is achieved when the simulated device behaves as a short circuit. Thus the potential of the Circuit boundary can be given by:

$$V_c = V_s - I \cdot R_s \quad (4.3)$$

4.2.2 Methodology

As mentioned in section 4.1.2, XOOPIC is capable of simulating plasma devices in both cartesian and cylindrical geometry. Hence an implementation was required for both geometry types. Thus, a general solution is first described, followed by t

General Solution

A case for the Circuit boundary shown in figure x. The points represent an arbitrary fixed grid, with grid points labelled i,j that correspond to the relative positions along the x and y

directions. The boundary itself is represented by the vertical line, placed along the grid points where $i=0$.

To determine the potential of the overall Circuit boundary, the potential at each grid point that intersects the boundary is first determined, then an average is taken. For a given point along the boundary, the potential can be computed using Gauss's law.

Gauss's law for a given pillbox, shown by the dashed line in figure x, is expressed as:

where Q is the charge enclosed by said pillbox. This charge can be expressed as a sum of the surface charge at the boundary and the charge density within the pillbox:

These two equations are combined, in equation x, and can be integrated with respect to their area and volume. This integration is depicted in equation x, resulting in an expression for the electric field at E_0 . Do note, that there are two areas produced: one related to the surface charge density (on the left); and one related to electric field out of the pillbox (on the right). The specific value of the area and volume terms depend on the geometry used, which is discussed later.

Because the electric field is proportional to the gradient of the potential, $E = -\nabla \phi$, a forward finite difference can be used:

As explained in section 3.x.x, the volume charge densities of particles are discretised onto the grid using bilinear interpolation, thus the charge density of the pillbox is simply its value at the grid ($\rho_{i,j}$). This leaves the surface charge density as the only unknown term on the RHS of equation x.

In order to determine the surface charge density, one can turn to Kirchoff's current law:

The conduction and convection current density can be expressed in terms of their charges, and backwards finite difference can be applied to give a new equation that describes the surface charge density at the current time step:

From this equation, $\hat{\sigma}_{t-1}$ and \hat{q}_{t-1} represent quantities that were determined in the previous time step; for the case where $t=0$, they can be initialised to zero. Q_{conv} represents the charge

deposited onto the boundary by particles since the last time step. Thus, the only unknown term on the RHS of equation x is \hat{Q}_t , which represents the conduction charge at the current time step.

This conduction charge can be determined using Kirchoff's voltage law (also expressed in equation x, though in a slightly different form):

The derivative term can again be solved numerically using backwards difference. However unlike the other equations, a second order difference is used here to provide a better approximation of the charge. This second order backwards difference is given as:

Combining the two equations x and x provides a new expression for the current conduction charge \hat{Q}_t :

Here, \hat{Q}_{t-1} and \hat{Q}_{t-2} are the charges at previous time steps (which can be initialised to zero), V_s and R are the source voltage and resistance respectively.

Using equation x, a new surface charge density expression can be made:

This in turn can be combined with equation x from Gauss's law to give a final solution of the potential at the Circuit boundary, seen in equation x. Note that the unknown terms (ϕ_0 and ϕ_1) have been rearranged to the LHS of the equation.

Typically, the coefficients in equation x can then be fed into field solver to determine the potential of all grid points. However in XOOPIC, the grid potentials of each boundary within the simulation are individually precomputed. Then for each time step, the overall potential is determined by summing the potential of individual boundaries via superposition, and adding the potential due to space charges. This approach does not necessarily speed up the computation of the field solver, however does decouple the code from boundaries from the field solver, making it easier to implement new boundaries.

Because of this precomputation, equation x can be simplified by specifying $\phi_1 = k_1 \phi_0$, where k_1 is simply the ratio of ϕ_1/ϕ_0 .

Cartesian Geometry

As alluded to above, the difference between the general solution in cartesian geometry apposed to the cylindrical geometry is the calculation of the area and volume terms: A_1 , A_2 , and V .

When operating in the cartesian coordinate system, the enclosed area of surface charge density and the area of the top of the pillbox are always equal, irregardless of the orientation. Thus the potential of Circuit boundary can be given as:

There are two possible orientations for the Circuit boundary for the cartesian geometry: along (or parallel to) the x-axis, or along the y-axis. These are illustrated in figures x and x respectively. The values for the area and volume in each orientation can be seen in table x.

Cylindrical Geometry

The two possible orientations for a cylindrical geometry are: along the r-axis, or along the z-axis. However, unlike the cartesian geometry where A_1 always equals A_2 for both orientation, this only holds for the case where the Circuit boundary is along the r-axis. For the case along the z-axis, A_1 and A_2 are distinct. This is illustrated in figure x.

The calculations to determine the area and volumes for each orientation is shown in table x.

4.2.3 Validation

The validation for the Circuit boundary was done in two parts: with a test case without particles and one with particles. The former was to evaluate the accuracy of the boundary calculations itself, without the computation for particle discretisation of the volume charge densities or the count of the number of particles that collided with the boundary producing convection charge. The latter was to test the operation of the boundary in a simulation scenario, which almost always has particles.

For both test cases, only simulation of a device in cartesian geometry is shown for the sake

brevity and the fact that solution to the potential of the circuit boundary is highly dependant on the geometry (as discussed in the previous subsection) thus are not directly comparable.

Test Case without Particles

An illustration of the device being simulated can be seen in figure x, where the Circuit boundary is placed along the y-axis at the left-most wall and a grounded *Conductor* boundary is placed along the right-most wall. This device, without any particles, essentially behaves as a capacitor with a capacitance given by:

Thus, this simulation models the charging of a capacitor in an RC circuit, which can be expressed as:

where the time constant is $\tau = RC$.

For the simulated device, $C = 8.85 \text{ pF}$ and $\tau = 8.85 \text{ ns}$. From this, an analytical solution can be determined, with the RC curve seen in figure x. For the XOOPIC simulation, a run was performed from $t=0$ to $t=50 \text{ ns}$ (approximately 5 time constants), with a time step of 10 ps . The resulting RC curve overlapped the the one produced by the analytical solution, hence only the absolute error is shown.

This error seen is a convergence error, which is due to the conduction charge values requiring a finite number of iterations before they converge to the true value. To illustrate this, the same simulation is run with a larger time step of 1 ns (100 times that of the original simulation). These results can be seen in figure x. Notice that there is a significantly larger absolute error (nearly 100 times greater). However looking at the percentage error, seen in figure x, the error values are quite comparable. (Though not shown here, running simulations for devices in cylindrical geometry reveal the same pattern, seen in figure x).

Test Case with Particles

To generate a simplified test case to only asses the impact of particles on the Circuit boundary was slightly more challenging. Nonetheless, the simulated for this device for this test case can be seen in figure x.

Notice that the grounded Conductor boundary has been replaced with a *Beam Emitter* boundary. The Beam Emitter is similar to the Current Source boundary, wherein a steady supply of charge is supplied to the boundary. However, unlike the Current Source which simply deposits the charge at the boundary which in turn generates a potential, the Beam Emitter releases this charge into the simulation domain in the form of particles. This steady source of particles is what enabled the evaluation of this test case. The particles emitted out of the Beam Emitter (i.e. to the LHS) in this case were electrons, hence why source current I_s is shown in the opposite direction. To ensure no particle losses, collisions by the MCC were turned off, the simulated device was set to a vacuum, and the drift velocity of the particles along the y-axis were set to 0 eV.

As for the Circuit boundary itself, the source voltage V_s was grounded, implying that the potential of the boundary is solely given by the current through the resistor, I_c . Additionally, because the source current must be equal the circuit current (as there were no losses) and one end of the resistor was grounded, the potential at the Circuit boundary has to be negative.

To analytical solution for this test case is given by the conduction current from electrons colliding with the Circuit boundary, and the displacement current across the capacitor. The conduction current is merely a step from 0 A to the -0.05 A specified by the source current; the negative sign is because current is leaving the boundary. The time at which this step occurs is given by the time taken for the electrons to travel across the device. As the drift velocity of the electrons along the direction of the x-axis is 10ev, thus its true velocity can be determined using the equation for kinetic energy:

This results in a velocity of approximately 1.87e6 m/s, hence it cross length of the device in 30 ps. Thus this is when the step in conduction current should occur.

As for the displacement current, this is given by:

Combining this with the analytical solution of the RC circuit in equation x, gives the displacement current of:

Akin to the conduction current, the displacement current at the surface of the Circuit boundary would only occur after the electrons colliding with the boundary itself.

The overall potential is obtained by combining the conduction and displacement currents with the resistance of the circuit. This is expressed in the equation below, and can be visualised in figure x.

For the simulation, the same overall device geometry was used as that of the test case without particles. Again, the simulations were run from $t=0$ to $t=50$ ns, with a time step of 10 ps. Figure x shows the absolute error of the Circuit boundary of the simulation as apposed to that of the analytical solution.

4.2.4 Future Work

Aside from the Circuit boundary simulating a more accurate “real world” voltage source, it can possibly be used in model more complicated circuit networks. This is because, Thevenin’s theorem dictates that any linear circuit network can be represented by an equivalent circuit that contains a voltage source with a series impedance (which is a resistance in this case).

Nonetheless, most circuit also contain some amount of capacitance and/or inductance. Hence, it could potentially be beneficial to add a series inductor and capacitor to the Circuit boundary in addition to the existing resistor. To achieve this, the expression for Kirchoff’s voltage law, equation x, needs to be modified to include the potential due to the inductor d^2Q/dtL and the potential due to a capacitor Q/C to the RHS of the equation. This new equation can be rearranged to solve for the current conduction charge. From there, the same process is applied to determine the potential of the Circuit boundary.

Chapter 5

Starter Simulations

Before diving into a complex device such as an electron gun, it would be useful to first simulate and then develop a simpler device. This allows the opportunity to utilise XOOPIC for long term simulations (not just test cases) and also experience to be gained by design and developing a device capable of generating plasmas.

The idea for this first experiment stemmed from a paper by Xu et al [1] where they utilised carbon-dioxide (CO_2) within a plasma reactor to harvest oxygen for epoxide. The plasma causes an oxygen atom from the CO_2 to disassociate, which in turn reacts to an alkene substrate that forms the epoxide. In the conclusion of the paper, Xu et al suggest that for this technique to be scalable the unreacted feedstock gas for the plasma should be recycled back into the system, which is precisely the experimental setup this chapter aims to perform.

The plasma used in the paper was Helium based, due to its inert property. The plasma was also generated at atmospheric pressure because the plasma-substrate interaction required to generate the epoxide is not compatible with vacuum pressures. This has the added benefit in terms of developing an experiment as atmospheric pressure devices have greater flexibility in terms of device geometry and fewer parts required as there isn't a need for pumps to generate and maintain the vacuum.

5.1 Split ring resonator

The aim is to ignite the plasma using a device known as a split ring resonator (SRR), seen in figure 5.1. The device consist of a conductive ring, usually copper, laid on top of a dielectric substance. Below the dielectric consist of a ground plane that covers the bottom of the device. Typically, a small gap is made on the copper ring that extends through the dielectric but not the ground plane; which is where the plasma discharge occurs. However since the goal of this experiment is to have a flow of gas through the plasma, the discharge gap needs to extend through the entire device.

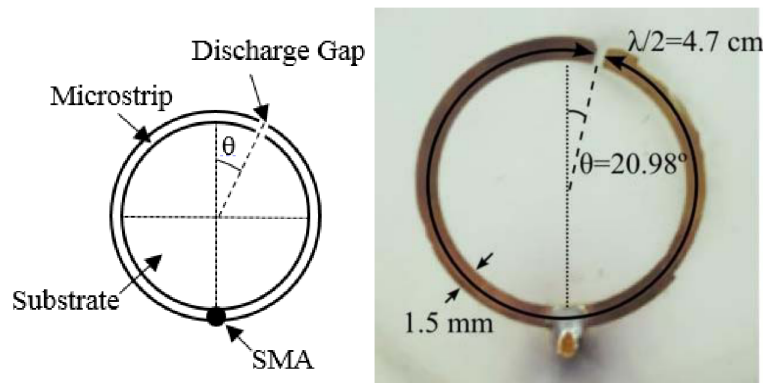


Figure 5.1: Schematic (left) and photo (right) of SRR [1].

To achieve a discharge, a microwave voltage is applied through the ring. The exact frequency to be used depends on the circumference from the middle of the ring. From this, the wavelength of the microwave input should be exactly twice the measured circumference. This in effect causes the ends of the SRR (where the gap is) to be 180° out of phase from each other. Because of this, when one end of the SRR is at the peak of the AC cycle, the other will be at a trough; thus the potential difference between the two ends has been doubled. This geometric trick allows a microplasma to form without the use of a lot of power.

Astute readers may notice another peculiarity with the SSR seen in figure 5.1, in that it is not symmetrical. Instead, the discharge gap appears to be offset towards one side of the device. This is deliberate as the offset gap allows for the impedance matching of the SRR to the impedance of the power supply used, thus maximising the power transfer. This offset angle is measured from the very centre of the ring, and is determined using the expression:

$$\theta = \arccos\left(1 - \frac{Z_{in}\pi}{Z_0 Q}\right) \quad (5.1)$$

where Q is the quality factor and Z_0 is the characteristic impedance of the SRR. Z_{in} is the input impedance of the power supply, which is typically 50Ω .

5.2 Simulations

When simulating in XOOPIC, the entire SRR is not simulated; instead a cross sectional plane of the discharge gap is modelled. A visualisation of this can be seen in figure 5.2. The reason for this is that the plasma formed is going to be constrained around the gap. Though the ring of the SRR is a circle, the discharge gap is quite small relative to the overall device, hence can be approximated to be a rectangle. Thus simulating a cross section of the gap would be valid assuming that the electric field across the gap is constant along all points of the ring.

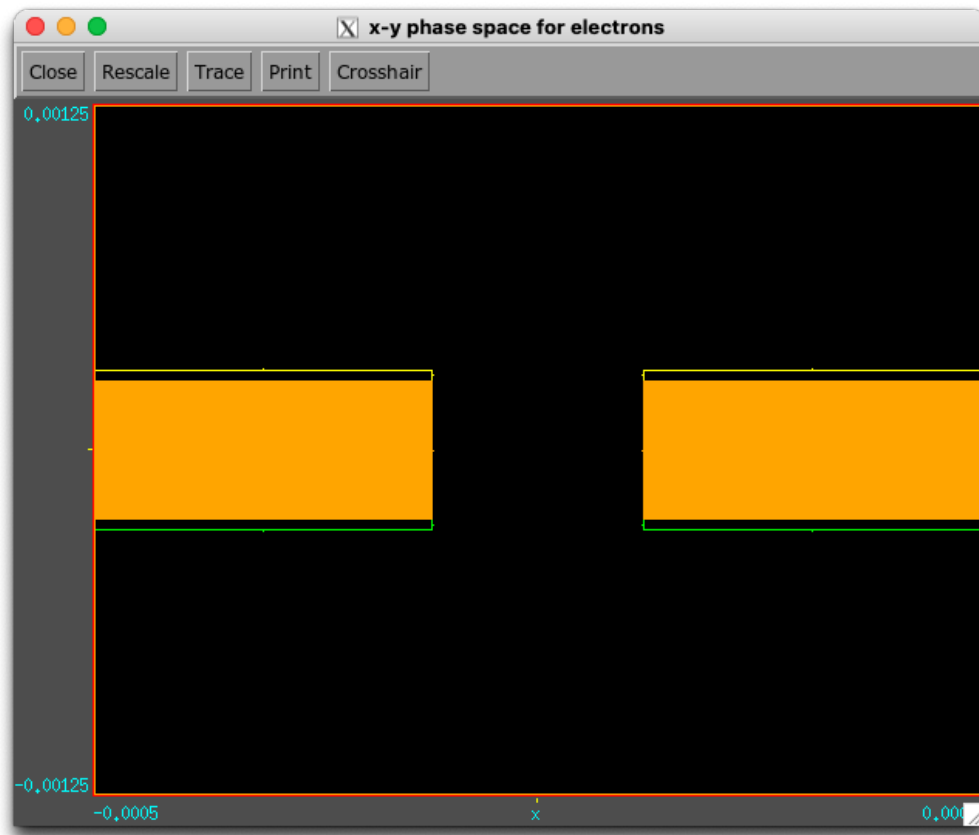


Figure 5.2: Cross section of SRR used for XOOPIC simulations.

5.2.1 Parameters

As for the input parameters for XOOPIC, the simulation domain was treated as a rectangle (in cartesian geometry) with a width and height of 1 mm and 2.5 mm respectively. The width was chosen as it was 1 mm is a standard inner diameter for PTFE tubes (which would provide the SRR with feed gas) whilst the height was an arbitrary selection as the tube would extend further. As such the left and right walls of the simulation were set to boundaries with a dielectric constant $\epsilon_r = 2.1$ whilst the top and top boundaries were treated as free space. Additionally, the entire region was filled with the He with a uniform density. The gas pressure was set to 1 atm and its temperature set to 25 meV (roughly equivalent to 290 K).

For the SRR within the simulation domain, the top of the device was set to an AC equipotential boundary, with the two ends being 180° out of phase, and a range of voltages tested between 110V to 180V. The bottom of the SRR was set to a grounded boundary. The thickness of all electrodes were set to 0.04 mm, being the standard for a PCB with 1 oz traces. The top and bottom electrodes were separated by a dielectric region. The exact parameters for this dielectric region are subject to change as its thickness and dielectric constant would affect the quality factor and thus the offset angle of the SRR. However, since a SRR device has not been build yet, the exact values required are unknown; hence the thickness was selected to be 0.5 mm with a dielectric constant $\epsilon_r = 10$. Finally for the discharge gap, a range of values were chosen between 120 μm to 480 μm .

In terms of the other parameters, a time step of 100 fs was used. The particle species being simulated included Helium ions and electrons with their standard masses and charges. MCC was turn on to facilitate collisions between the charged particle species and the background.

5.2.2 Results

Discharge Gap

The first set of simulation were run to evaluate four set of widths for the discharge gap: 120 μm , 240 μm , 360 μm , and 480 μm . For all three gaps, an AC voltage of 130 V was chosen. Figure 5.3 shows the number of ions produced over time, and the time series plot for the number of electrons produced can be seen in figure 5.4. One point of note for these graphs are its sawtooth nature where the number of ions and electrons increase before they sharply decrease in number. This is a feature of XOOPIC that halves the total number of particles when a certain particle limit has been reached, in order improve the simulation performance.

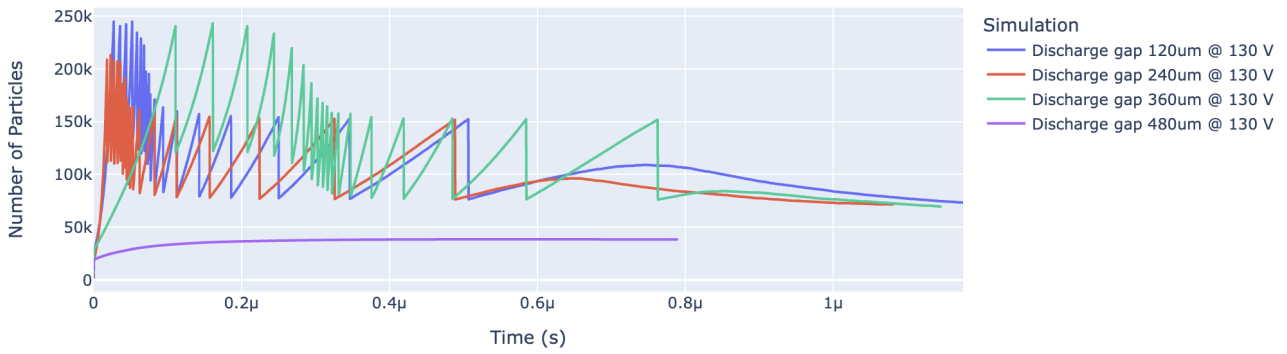


Figure 5.3: Time series plot of ions across different discharge gap widths.

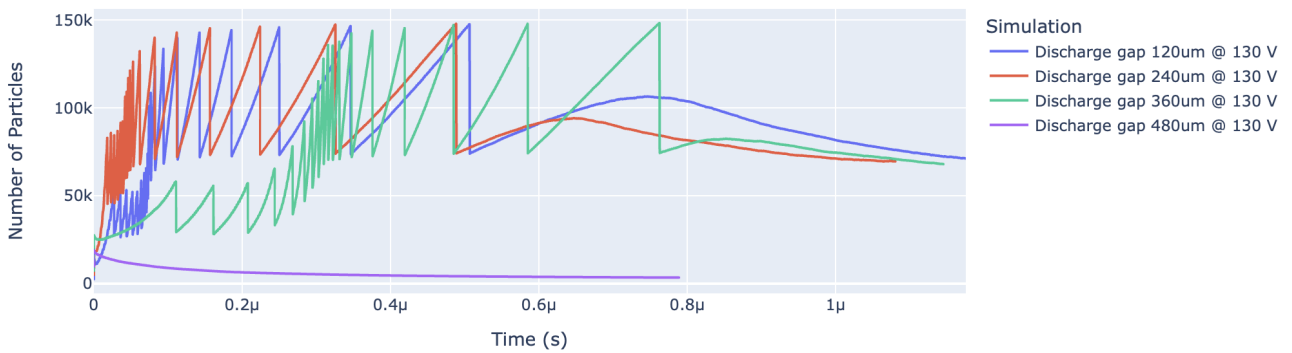


Figure 5.4: Time series plot of electrons across different discharge gap widths.

From the four tests, only three show a growth in the number of both ions and electrons. The simulation with a discharge gap of 480 μm did show an initial growth in ions however as all the electrons were used up, that number plateaued. Because of this the simulation was terminated

earlier hence the abrupt stop in data. As for the other three tests, the gap of 360 μm performed the worst in terms of growth and the gap of 240 μm performing the best.

Applied Voltage

Based on the first test of the gap widths, a second set of simulations were run assessing different voltages: 110 V, 130 V (from previous tests), 150 V, and 180 V.

Bibliography