Prediction of Non-Equilibrium Gas Filling and Ionization Processes in a Chamber Using Direct Simulation Monte Carlo (DSMC) Methods

by

Benjamin Humer

A thesis submitted in conformity with the requirements
for the degree of BASc in Engineering Science

Department of Applied Science and Engineering
University of Toronto

Prediction of Non-Equilibrium Gas Filling and Ionization Processes in a Chamber Using Direct Simulation Monte Carlo (DSMC) Methods

Benjamin Humer
BASc in Engineering Science

Department of Applied Science and Engineering
University of Toronto
2024

Abstract

This undergraduate thesis investigates the neutral gas distribution over time during the vacuum filling process in General Fusion's PI-3 plasma injector. Previously, General Fusion had not simulated the dynamics of the PI-3 filling process, so there is an opportunity to further understand the gas distribution at the time of ionization, potentially increasing the total amount of plasma produced by modifying the filling process. This research models the gas distribution within an approximated PI-3 geometry over time using Direct Simulation Monte Carlo methods, and includes a detailed exploration of collisional Particle in Cell codes that could be used to extend this work into a full ionization model. The final simulations of the PI-3 plasma injector showed promising particle density in the Penning trap with some accumulation at the tip of the ionization chamber as well. It also showed that the peak number density was reached after over $300\mu s$ despite General Fusion only running their gas puff for $240\mu s$. This should be further explored in the ionization model and other experiments to further optimize plasma formation in the PI-3 plasma injector.

Acknowledgements

First, I want to thank Clinton Groth the primary supervisor of this project. He chose to take on an additional project from outside his research group, and without his guidance I would not have been able to complete this project to the level that I did.

Secondly, I want to thank Abetheran Antony and Ryan Zindler of General Fusion. Abetheran helped supervise the project from the corporate side, and Ryan was the person who originally coordinated the project so this couldn't have happened without both of their help.

Finally, I want to thank SciNet. They were great at providing support and helping troubleshoot issues I had, and without their computers I would not have been able to run my final DSMC simulations at higher resolution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Description and Background

This thesis project involves the use of Direct Simulation Monte Carlo (DSMC) methods to predict non-equilibrium gas filling in a small chamber, and then researching how this simulation could be extended to include ionization as well (see Chapter 2 for more information about General Fusion and their approach to fusion energy Generation). The project and problem definition were provided by General Fusion [1], but will be primarily supervised by Prof. Clinton Groth [2] from the University of Toronto Institute of Aerospace Studies (UTAIS) with assistance from Abetheran Antony who works in Applied Physics at General Fusion. General Fusion's approach to fusion energy is a hybrid of Inertial and Magnetic Confinement where the plasma is formed in their Fusion Plasma Injector [3] (a form of Plasma Railgun similar to the one proposed by John Marshall [4]) and then fired into a central chamber to undergo fusion. This plasma injection process starts with gas flowing into an evacuated chamber from a number of different valves; then before equilibrium is reached, that gas is ionized with extremely high voltage and compressed using the Fusion Plasma Injector. The beginning stages of this process are what General Fusion is interested in learning more about through the work conducted in this thesis. Currently General Fusion assumes a distribution for the diffusion of gas in the chamber, and then models the ionization and plasma injection from that point, but this can be made more accurate with DSMC simulation of the initial gas diffusion.

1.2 Proposed Gap In Existing Research

The proposed research gap is to properly understand the density distribution of neutral gas in the PI-3 Ionization chamber. Up to this point, the assumed distribution of the gases in the chamber was a Gaussian about each valve. General Fusion doesn't have any existing simulations that they

have used to determine the process of filling in the PI-3 ionization chamber, so any insight that can be gained from the DSMC simulations in this thesis will fill that gap in knowledge.

1.3 Central Research Objective

The objective of this thesis work is to provide a more accurate description of the initial gas distribution at the time of ionization, as well as the distribution of ionized gas before compression. Since it is based on DSMC, this description will incorporate all the relevant particle interactions so that General Fusion can learn more about where the gas is when ionization occurs in their plasma injector [5] and where most of the ions are distributed within the chamber before compression. DSMC should be a great tool for tackling this problem provided the Knudsen Number (see Section 3.1.1) and flow regime (see section 3.1.2) make it computationally viable. This is because DSMC simulates the fundamental physical interactions and is great for problems like this without known solutions.

1.4 Research Approach

The approach that was taken was modeling the neutral gas distribution through DSMC simulation. DSMC simulation is a very accurate and comprehensive modelling tool for neutral gases in non-equilibrium systems (which is what will initially be simulated) [6]. Sections 3.1, 3.2, and 3.2.4 explain the fundamentals of DSMC, while Sections 3.3 and 3.4 go over a variety of different open source codes that can be used to perform this type of neutral gas simulation.

Once the neutral gas distribution was simulated, the project was extended to include a detailed literature review on the current state of collisional PIC simulations (see Chapter 4 for more details). This extension sets the foundation for continuation of this project, and would allow for an ionization model to be quickly built onto the current neutral gas filling results.

1.5 Research Significance

General Fusion is looking to develop technology that can produce large amounts of carbon free energy with minimal radiative byproducts. More information about the way gas is distributed in the chamber at the time of ionization would allow for more control of the plasma in the injector itself. This should lead to a higher number of fusion events occurring in the plasma, moving General Fusion one step closer to net energy production. This research also advances General Fusion's arc ionization model, which is applicable to a variety of different fusion systems. Since it will be developed based on open source code, the arc ionization model would also remain open source, and there are very few existing models of this type which means it could be useful to others beyond General Fusion.

Chapter 2

Background: General Fusion

2.1 General Fusion

General Fusion is a Canadian company based in Vancouver B.C. trying to develop a fusion power device using the concept of Magnetized Target Fusion (MTF) [1] (see Section 2.2 for more information). Since being founded in 2002, the company has built a number of prototype fusion devices to demonstrate various concepts that will be necessary for use in their final MTF reactor. This lead to the construction of the PI-3 Plasma Injector in 2017, which is a full reactor scale plasma injector (see Section 2.3 for more information). The PI-3 device is the subject of the simulations in this thesis.

2.2 Magnetized Target Fusion

Magnetized target fusion is a concept that combines features of both magnetic confinement fusion (MCF) [7] and inertial confinement fusion (ICF) [8]. These approaches are combined by utilizing magnetic fields to confine the fuel while it is at lower densities (like MCF) during the plasma formation process, and then using rapid compression to greatly increase the fuel density and temperature of the plasma once it is formed (like ICF). This leads to a lower density than traditional ICF, but the integration of MCF concepts will allow for longer confinement times and superior heat retention when compared to ICF experiments [1]. ICF experiments using lasers to compress their fusion fuel into plasmas have been able to generate net thermal energy (not considering losses in the broader system and fuel preparation) [9], so enhancements to confinement time could increase the efficiency of this process and bring it to a commercially viable scale.

2.3 PI-3 Plasma Injector

PI-3 was built by General Fusion in 2017 for the purpose of studying the physics of a reactor scale MTF plasma injector. It is a large, single stage Marshall Gun [4] that injects plasma into a spherical tokamak, which provides greater thermal confinement than the spheromak that was used as the reaction chamber of the PI-1 and PI-2 injectors. The PI-3 plasma injector operates in three distinct stages, which are listed below and labeled on Figure 2.1.

1. Neutral Gas Filling
2. Neutral Gas Ionization
3. Plasma Transport

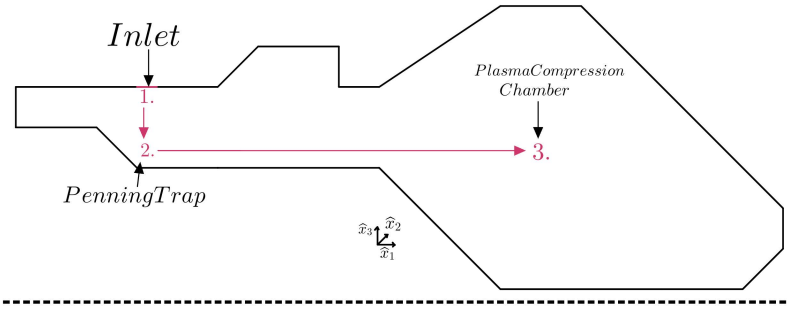


Figure 2.1: Labeled axisymmetric diagram of the three operational stages of the PI-3 plasma injector, and how particles move between these stages. The line of symmetry is denoted by a dashed line on this figure.

2.3.1 Neutral Gas Filling

Before operation, PI-3 is evacuated using a vacuum pump to a pressure of $P = 7 \cdot 10^{-4} Pa$, and a corresponding number density of $N = 2 \cdot 10^{17} m^{-3}$. Number density is more convenient to work with throughout this thesis due to the ease of calculating other flow related parameters (see Section 3.1) and relating values to the final results (found in Section 6.5). The number density and the pressure are related by Equation 2.1, which is derived from ideal gas law.

$$PV = NRT \Longrightarrow \frac{N}{V} = n = \frac{P}{RT} \tag{2.1}$$

After reaching its evacuated state, the chamber is filled through 25 equally spaced valves that are placed around the edge of the chamber (labeled as point 1. in Figure 2.1). This inflow process is what this thesis primarily addresses.

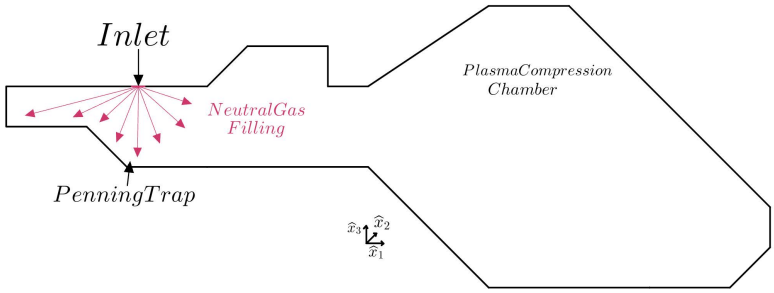


Figure 2.2: Labeled axisymmetric diagram of the PI-3 plasma injector. The dynamics of the first operational stage are sketched on the diagram.

2.3.2 Neutral Gas Ionization

After being filled, the gas is at a high enough density that it can be ionized, and can cause significant electron impact ionization leading to a large ionization cascade throughout the neutral gas. This ionization is due to the applied electromagnetic fields shown in Figure 2.3, and most of the ionization takes place within the Penning trap (labeled as point 2. in Figure 2.1). More information about the Penning trap that General Fusion uses is available in Section 4.1.2.

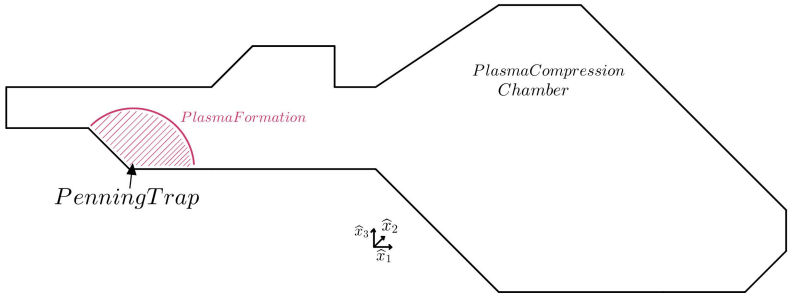


Figure 2.3: Labeled axisymmetric diagram of the PI-3 plasma injector. The dynamics of the second operational stage are sketched on the diagram.

2.3.3 Plasma Transport

After being formed in the Penning trap, the electromagnetic fields change from those shown in Figure 2.3 to those shown in Figure 2.4. This causes the plasma to accelerate from the Penning trap towards the wider part of the reaction chamber where it would be compressed in a fully functioning MTF reactor be compressed causing the ions to fuse together. This is where a PI-3 experiment ends, with the plasma cooling down in what would normally be the compression chamber.

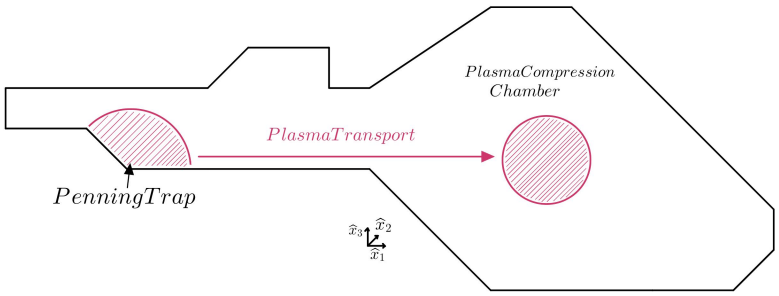


Figure 2.4: Labeled axisymmetric diagram of the PI-3 plasma injector. The dynamics of the third operational stage are sketched on the diagram.

Chapter 3

Literature Review: DSMC

3.1 Fluid Mechanics Background for Neutral Gases

In this section, theory and background information is presented for technical readers who are not specialists in fluid mechanics or gas dynamics. Specialists in fluid mechanics or gas dynamics may not need to read Sections 3.1 of the report, as it will be referenced in further sections of the report so readers can return to this technical content if required. Those familiar with the intricacies of DSMC may also not need to read Section 3.2.

3.1.1 Knudsen Number

Many key question associated with this thesis revolve around why DSMC is well suited for tackling this specific type of gas dynamics problem, and the foundation of this discussion depends on an understanding of flow regimes. The flow regimes covered by present theory are often classified by their Knudsen Number.

The Knudsen Number is a dimensionless number defined as the ratio of the molecular mean free path to the representative physical length scale (see Equation 3.1).

$$Kn = \lambda / L \tag{3.1}$$

Where λ is the molecular mean free path (see [10] for a basic definition) in the medium of interest, and L is some representative physical length scale such as the gap length that a gas could travel through before collision with a solid surface. This can intuitively be understood as a metric for how collisional the medium will be by thinking about the edge cases of the values the Knudsen Number can take on, so from Equation 3.1 we take the following 2 cases.

1. $Kn \to \infty$: then the mean free path is infinitely larger than the length scale, and molecular

collisions have very little impact on the evolution of the system.

2. $Kn \to 0$: then the length scale is infinitely larger than the mean free path, statistical fluctuations disappear since the molecules of the fluid are effectively constantly colliding, and are effectively homogeneous at a small enough length scale.

In case 1, statistical fluctuations will be dominant for a fine resolution hexahedral grid (where $V \approx L^3$ for the grid). This is because within each cell, there will be no collisions, resulting in free diffusion based on the statistical profile of the gas in each cell.

In case 2, the cancellation of statistical fluctuations leads to the functions describing the gas at different points in space being continuous. Consider an example function such as $T(\vec{r}, t)$ representing the temperature of a fluid at various points in space $\vec{r}$ at times t. For an infinitely collisional gas, properties of the gas in two infinitesimally nearby locations are going to be the same, since they will exchange energy and momentum through their frequent collisions. This leads to the equality in Equation 3.2 seen below.

$$T(\vec{r}, t) = \lim_{d\vec{x} \to 0} T(\vec{r} \pm d\vec{x}, t) \tag{3.2}$$

The difference in modelling these 2 extreme cases comes from these continuous functions seen in case 2. In case 2, the model can be abstracted to these continuous properties and the effects they have on each other; while in case 1 the dynamics of individual molecules or groups of molecules must be modelled for the full physical picture to be understood. These regimes and the different models that apply to them are further discussed in Section 3.1.2.

3.1.2 Classification of Flow Regimes

With the knowledge of Knudsen Number presented in Section 3.1.1, gas dynamics processes can be classified based on their Knudsen Number. These classifications are separated into four basic categories presented in Figure 3.1, where the molecular regime, and continuum regime are recognizable from Section 3.1.1 where they were labeled cases 1 and 2 respectively.

With an understanding of the 2 extreme cases presented in Section 3.1.1, understanding can be extended into the slip flow and transition regimes shown in Figure 3.1. In the molecular regime, the intuition formed in case 1 fully applies, and in the continuum regime, the intuition formed in case 2 fully applies. The continuum regime will not be discussed further, but more information is available in the following set of notes [12]. In the slip flow regime, the internal flow can be thought of as the same as in continuum flow, but on the edges there may be different phenomena based on the imposition of boundary conditions on the flow (see Section 4 of [13] for more details about flow boundary layers). In the transition regime, there is a mix of both free path DSMC particles as well as sufficient collisional frequency to preserve continuum like trends in the data, but depending

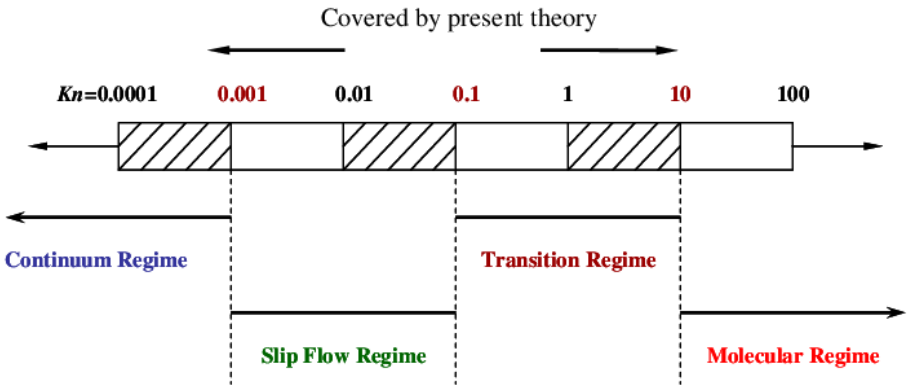


Figure 3.1: Flow regimes covered by present theory classified based on Knudsen Number [11].

on the Knudsen Number, these statistical fluctuations keep continuum models from being accurate. The different fluid flow models in terms of their flow regimes are described in Figure 3.2.

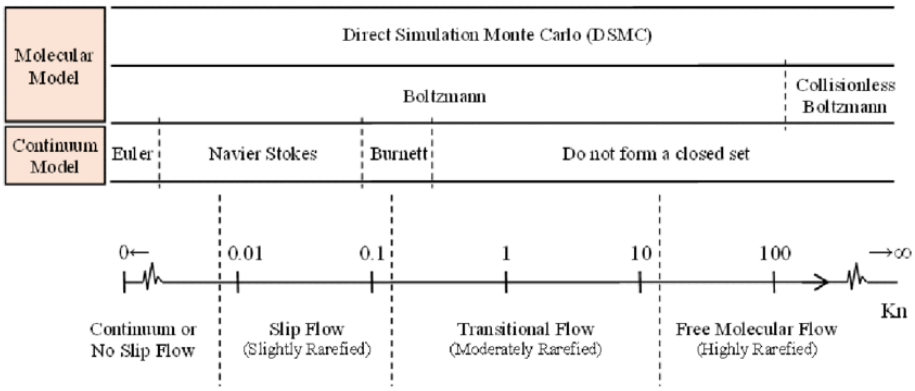


Figure 3.2: Continuum mechanics simulation types classified based on the Knudsen Number regimes over which they provide accurate numerical results [14].

From Figure 3.2 (see [14] for further detail), we can see that the continuum models for a fluid flow are valid until the beginning of the transitional flow regime, but after that the end of the Burnett Hydrodynamics Equations, there is no closed set of equations that can describe the flow of a fluid. When considering molecular models however, there is no Knudsen Number range in which they do not provide accurate solutions. This is because they are simulating the fundamental physical interactions that govern macroscopic properties of interest in continuum simulations (see case 2 in Section 3.1.1 for an example). This means that for a simulation that begins with part of the domain having a Knudsen Number of $Kn < 0.001$, and another part of the domain having a Knudsen Number of $Kn \to \infty$, and a final Knudsen number of $Kn > 0.1$, then modelling the entire domain using a molecular model that accounts for collisions (such as a Boltzmann or DSMC model) is perfectly valid. The flow regimes for the specific gas filling process that General Fusion is interested in are analyzed in Section 5.2.3.

3.2 Direct Simulation Monte Carlo (DSMC)

Direct Simulation Monte Carlo (DSMC) is a method for simulating molecular dynamics in fluids of all Knudsen Number regimes, but is largely used for highly rarefied flows [15].

3.2.1 Key Assumptions of DSMC

The following four assumptions about the physical characteristics of the system are utilized in DSMC [15].

1. Molecules move without interaction in free flight for the length of the chosen time step [15].
2. Impact parameters and initial orientations of colliding particles are random [15].
3. In every cubic mean free path there are many molecules, but only a small fraction need to be simulated for the molecular description of the flow to be accurate [15].
4. The only molecules that any molecule could collide with are the molecules within a molecule's finishing cell.

For dilute gases, all four assumptions can be made so they are very accurate with the right parameters. This makes it so the DSMC method produces accurate simulations of macroscopic non-equilibrium flow fields. For dilute gases, there is always a time step which allows assumption 1 to be true regardless of Knudsen Number; Assumption 2 is true provided there is sufficient molecular spacing; assumption 3 is the true foundation of DSMC when compared to Collisional Boltzmann, and can be achieved using a number of different computational methods described later on in this section; and assumption 4 can always be made true with a sufficiently small time step just like assumption 1.

From the four above assumptions, we can understand the basics of a DSMC program (the DSMC algorithm is described in more detail in Section 3.2.4). Just like in other molecular dynamics simulations, the state of the system is determined by a list of properties associated with each individual particle. In DSMC, these properties must include at least the position and velocity ($\vec{x}(t)$ and $\vec{v}(t)$) of each particle at any time step t of the simulation, but additional quantities such as internal rotational energy of the particle ($E_{rot}(t)$) are also quite common to more accurately handle energy and momentum transfer in complex collisions.

3.2.2 How DSMC Differs from Other Molecular Dynamics Simulations

The way DSMC differs from other molecular dynamics simulations is by not simulating all the particles in a given domain, just a *representative sample* of the particles, which was alluded to in

assumption 3 from earlier in this section (Section 3.2.1). In any given time step Δt where $\Delta t = t_f - t_i$, the particle moves from its specified position ($\vec{x}(t_i)$) to a final position by following its previous velocity trajectory ($\vec{x}(t_f) = \vec{x}(t_i) + \vec{v}(t_i) \cdot \Delta t$). Then based on the path length travelled over the time step ($d = \vec{v}(t_i) \cdot \Delta t$), a probability of collision is assigned to each particle, and collisions with other particles within their cells are simulated probabilistically regardless of whether their paths actually crossed. This is because if each cell has a representative set of particles, then a collision with any particle from the representative set at any random angle is the best approximation for modelling all the collisions with the highest degree of computational efficiency.

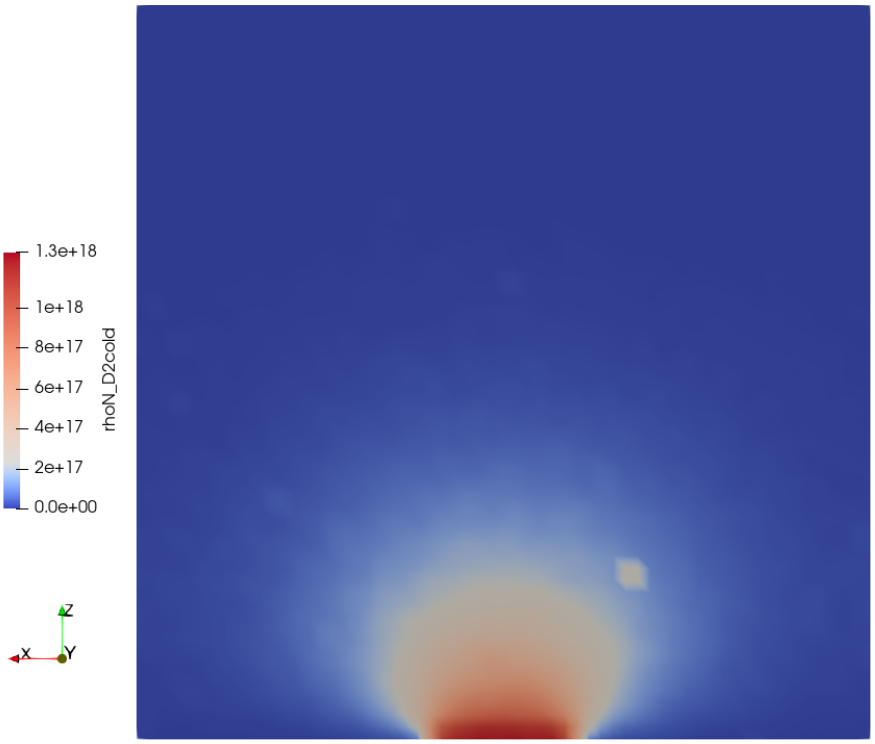


Figure 3.3: Cube with inflow patch in a slit on the center of the $z = 0$ plane partway through a filling simulation.

3.2.3 Why Do Molecules Need to stay in Their Cells for Multiple Time Steps

The fourth assumption from Section 3.2.1 does not appear in [15], but it is important for the specific case of a transient gas filling simulation, which is the core problem of this thesis. In a gas filling process from an inlet, as gas fills the chamber there will be a bias to the faster end of the velocity distribution (shown in Figure 3.4). This is due to the initial condition of the density, which is a step function where there is a high density area of gas that fills into a near vacuum (a snapshot

of the density in a square chamber partway through the filling process is shown in Figure 3.3). The total flux through the inflow surface is described by Equation 3.3, where n is the number density of the gas in $[molecules/m^3]$, and c_{gas} is the average velocity of the gas inflow in $[m/s]$. The equation for c_{gas} can be derived from the Boltzmann Distribution, and it is shown in Equation 3.4.

$$\Gamma = \frac{1}{4} n c_{gas} \tag{3.3}$$

$$c_{gas} = \left(\frac{8kT}{m\pi} \right)^{\frac{1}{2}} \tag{3.4}$$

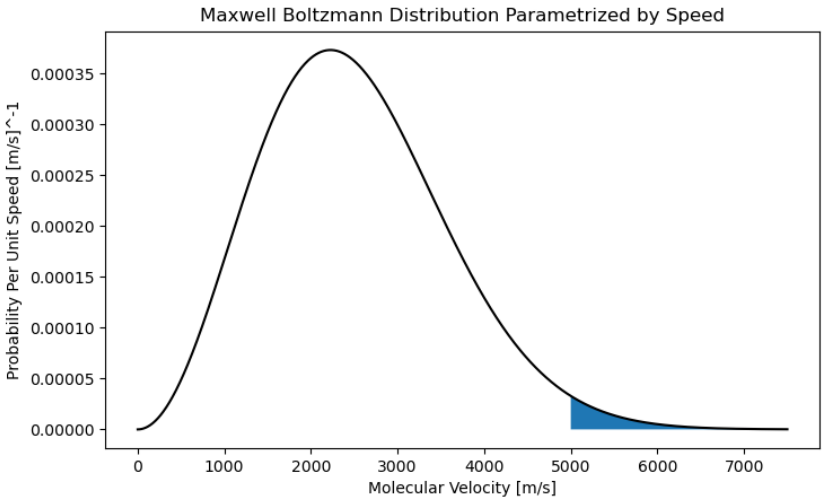


Figure 3.4: Plot of a Maxwell Boltzmann Distribution at $T = 300K$, parameterized on molecular velocity with the high energy section of the distribution highlighted.

This is important for filling, because the particles at the high velocity end of the Maxwell Boltzmann Distribution naturally form the leading edge of the gas (the leading edge of the gas is very visible in Figure 3.3 as the lightest part of the flow expanding outwards) from a density step function initial condition, so the gas is not thermalized; and as a result the distribution of energies in each cell will be different. This means the particles must not cross through multiple cells in one time step, otherwise the energy transfer between particles will not be representative of true physical phenomena. If particles pass through multiple cells over the course of one time step, then they do not properly represent the collisional probabilities in the cells they pass through along their path, just the cell they end up in at the end (see Figure 3.5).

In Figure 3.5, a particle moves along the entirety of $\vec{d}$ in one time step. This is not a problem if properties of cells A, B, and C (such as energy distribution of particles, vector velocities of particles, and the macroscopic number density) are the same, but for a transient problem, adjacent cells will have different properties across the simulation domain. This means if a particle doesn't spend mul-

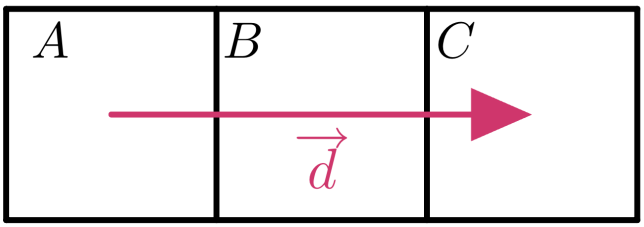


Figure 3.5: Diagram showing a particle moving from cell A to cell C of a grid in one time step, skipping over cell B.

tiple time steps in each cell, it is not properly accounting for the different collision probabilities and energy transfer parameters from the different cells, leading to incorrect simulation of the underlying physics.

3.2.4 The DSMC Algorithm

In a DSMC program, the program runs in a loop from initial time t_i to final time t_f in N time steps of length Δt specified by Equation 3.5. At each time step, the following list of steps occur in the specified order.

$$\Delta t = \frac{t_f - t_i}{N} \tag{3.5}$$

1. Generate particles at inflow boundaries (or through imposition of initial conditions at $t = 0$) [15].
2. Move all particles in straight lines along their molecular velocity vectors for a time step less than the local mean collision time [15].
3. Apply Boundary Conditions to particles where it is appropriate (wall collisions and particle deletions) [15]
4. Perform stochastic collisions within each cell, assuming parameters for each collision pair to be random [15].
5. Sample particle properties in each cell [15].

Particle Generation

At the beginning of each time step, particles are generated at any inflow patches. They are randomly assigned parameters, usually based on a modified Maxwell Boltzmann Distribution parameterized based on Temperature, and often include the ability to manually bias the mean velocity

by some vector $\vec{v}_{bias}$. The Maxwell Boltzmann distribution for 3-dimensional velocity is shown in Equation 3.6.

$$f(\vec{v}) \equiv \left[\frac{2\pi kT}{m}\right]^{-\frac{3}{2}} exp\left(-\frac{1}{2}\frac{m\vec{v}^2}{kT}\right) \tag{3.6}$$

The modified distribution mentioned in the previous paragraph takes the velocity $\vec{v}$ from Equation 3.6 and then adds it to $\vec{v}_{bias}$ as is done in Equation 3.7.

$$\vec{v}_{particle} = \vec{v} + \vec{v}_{bias} \tag{3.7}$$

Particle Movement

Particles created at the inflow patch were randomly assigned a position $\vec{x}(t_i)$ somewhere on the face of the inflow patch in the previous step; and particles that were already part of the simulation for the previous time step have positions $\vec{x}(t_i)$ which were saved to represent the system state after the previous time step.

At every time step, all particles (both created at the inflow patch and existing in the domain) are moved by a distance d along $\vec{v}(t_i)$ (specified by Equation 3.8).

$$\vec{d} = \vec{v}(t_i) \cdot \Delta t \tag{3.8}$$

This moves each particle to a position $\vec{x}(t_f)$ which is defined in Equation 3.9.

$$\vec{x}(t_f) = \vec{x}(t_i) + \vec{v}(t_i) \cdot \Delta t = \vec{x}(t_i) + \vec{d} \tag{3.9}$$

After this movement of particles, the set of positions $\vec{x}(t_f)$ of the particles forms the positions $\vec{x}(t_i)$ of the next time step. This means the position state of the system has been fully specified at this point.

Boundary Conditions

Now that particles have been moved to their final positions, boundary conditions need to be imposed on the particles that have left the domain. The main two boundary conditions that will be used throughout this thesis are wall boundary conditions and deletion boundary conditions. Particles that have left the domain and would have collided with the wall are brought back to the point of contact, and then collide with the wall in whichever way is specified by the boundary (see [16] for some examples). If a particle that has left the domain would have collided with a deletion patch, then the particle is removed from the state of the simulation so it is not a part of the simulation at the beginning of the next time step. Now only the particles that are actually entering the next time

step of the simulation remain as a part of the saved state.

Stochastic Collisions

In a gas composed of hard sphere molecules of diameter d (twice the total atomic radius of each molecule), the probability that two molecules both located within a volume of gas V (the volume of the cell in which both molecules are contained) collide is proportional to the total volume swept out by their interaction cross sections normalized by the volume of gas being considered [15]. This is shown in Equation 3.10, and the concept of volume swept out by an interaction cross section [15] can be visualized in Figure 3.6.

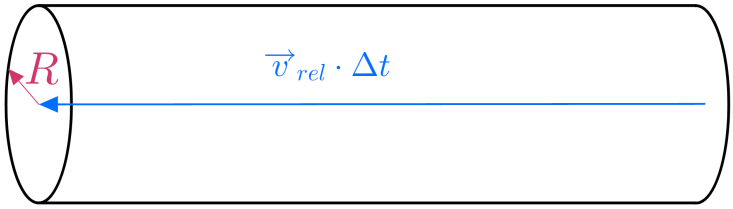


Figure 3.6: Diagram showing how the interaction cross section sweep over time can be represented by a volume element.

In Figure 3.6, the total volume swept out by the interaction cross sections of the two molecules is simplified to the case where one of the two molecules is at rest [15]. This is always able to be the case for any two particles by choosing v_{rel} as the relative velocity of one of the molecules, so then the other is always at rest in that reference frame. Then the interaction cross section forms the area that is traced along the path length of the moving molecule, forming a cylinder. This can be used to understand Equation 3.10.

$$P \propto \pi d^2 v_{rel} \Delta t / V \tag{3.10}$$

In Equation 3.10, it is defined that two molecules both exist within a volume V, then their total interaction volume is the volume of the interaction cylinder presented in the previous paragraph, so the ratio of these two areas is the interaction probability of any two real particles. However, in DSMC, each *DSMC Particle* represents a large number (W_p) of real particles which all have the same velocity and molecular properties as the DSMC particle [15]. This allows us to find the probability of a DSMC interaction by adding a factor of (W_p) to Equation 3.10.

$$P_{DSMC} = \pi d^2 W_p v_{rel} \Delta t_{DSMC} / V_{DSMC} \tag{3.11}$$

Then since in each cell there are $N_p(N_p - 1)/2$ possible particle pairings, P_{DSMC} from Equation

4.9 should be multiplied by all possible particle pairings to find the total number of collisions that occur in a cell.

$$N_{coll} = \frac{1}{2} N_p (N_p - 1) \frac{\pi d^2 W_p \langle v_{rel} \rangle \Delta t_{DSMC}}{V_{DSMC}} \tag{3.12}$$

Where the only change from Equation 4.9 to 3.12 is that the relative velocity is now replaces by $\langle v_{rel} \rangle$, the mean relative velocities of all the combinations.

Now collisions between the representative particles can be simulated since Equation 3.12 shows how many would be expected to occur, and these collisions determine the velocity states of the particles. Now we have the full velocity state of the particles, as well as the internal molecular states if those are a part of the simulation.

At this point, there are further pieces of mathematics that are used to increase the computational efficiency of the code, but this contains the main physical intuition required to understand the work done in this thesis. If the specifics are of interest, Section 6.2.3 of [15] is a great resource to begin with.

Sampling Cell Properties

Although DSMC simulations are molecular in nature, often the properties of interest are macroscopic, as they are in the case of this thesis. This means at each time step, the particles can be analyzed to give macroscopic properties in each cell (such as the number density in each cell) should they be desired to be saved for further analysis.

3.3 Existing Open Source DSMC Packages

This section outlines various modern DSMC packages that have been designed for creating numerical solutions to highly rarefied gas dynamics problems. They are listed in chronological order of first release, and the first line of each entry shows that release date, along with the most recent update, where *present* means that there are more features continually being added.

3.3.1 Graeme Bird

1963-2013 [17]**:** *Graeme Bird Website*

Graeme Bird proposed and developed the DSMC method, and continued to be at the forefront of the field until he retired in 2013 at the age of 88 [17]. His software is very user friendly, but I've found that the open source versions are more designed for demonstrating innovations in DSMC (such as new collisional models). The trade off for these user friendly options is that the source code is unavailable, so new features cannot be added by users.

3.3.2 OpenFOAM

2009-present [18]**:** *OpenFOAM Website*

OpenFOAM is a free open source CFD software package developed by OpenCFD for use in many different areas of engineering and science [19]. Development began in 2004, but the DSMC package was first available in 2009. The biggest strength of OpenFOAM is strict coding practices and the dictionary and class structure for specifying program parameters. This (along with the source code being open source) allows for OpenFOAM users to add their own features.

3.3.3 hyStrath

2014-present [20]**:** *hyStrath Website*

The GitHub repository *hyStrath* is an OpenFOAM extension that features hypersonic and rarefied gas dynamics code [20]. The code is specifically designed for atmospheric re-entry analysis, but is suitable for all kinds of rarefied gas dynamics problems, and has a module called dsmcFoam+ which is an extension of OpenFOAM's DSMC software. It is also built within the OpenFOAM framework and strictly adheres to OpenFOAM's coding practices, so it is easy to expand their existing code base to add additional features.

3.3.4 SPARTA

2014-present [21]**:** *SPARTA Website*

SPARTA is an acronym for Stochastic PArallel Rarefied-gas Time-accurate Analyzer [21]. It is a parallel DSMC code for low density 2-D or 3-D gas simulations created at the Sandia National Laboratory in the United States [21]. The source code is available, but the coding practices aren't standardized, and there is limited documentation.

3.3.5 Starfish

2016-present [22]**:** *Starfish Website*

Starfish is a 2-D gas and plasma simulation code [22] developed by Particle In Cell Consulting (PIC-C). For modelling plasma, it uses the Electrostatic Particle in Cell method; and for modelling free molecular flow it can use several continuum solvers as well as DSMC. There are two versions. One is free and open source but contains limited features, and the other is only internally available within PIC-C, but can be requested in a compiled format.

3.4 Comparison of Existing Open Source DSMC Codes

The DSMC codes were evaluated in a simple comparison matrix which is shown in Table 3.1. The results of this comparison matrix were evaluated holistically through a series of discussions with the supervisor of this thesis at General Fusion, Abetheran Antony. The evaluation was centred around what would be the best way for General Fusion to get as much information as possible out of the thesis while also prioritizing my learning and educational development. We decided on hyStrath because it has a great list of features and has been thoroughly verified in academic papers [23].

Table 3.1: Comparison matrix for evaluation of different Open Source DSMC codes.

Feature	**Bird**	**OpenFOAM**	**hyStrath**	**SPARTA**	**Starfish**
2-D	Yes	Yes	Yes	Yes	Yes
3-D	Yes	Yes	Yes	Yes	No
Axisymmetric	Yes	No	Yes	No	Yes
Built in Ionization	No	No	No	No	Yes
Mesh Generation Utility	No	Yes	Yes	No	No
Documentation /10	10	7	6	4	7
Supporting Literature /10	10	7	7	6	5

Chapter 4

Literature Review: PIC

4.1 Electromagnetism Background for Dynamic Systems

Particle in Cell (PIC) simulations are conceptually similar to DSMC simulations (see Section 3.2 for more information) since they also use a model of 'equivalent particles' to form representative samples that can then be used to determine macroscopic quantities. Where PIC and DSMC approaches differ is the way that these particles are moved. In DSMC, the particles are moved based on their current trajectory and that trajectory is only modified by collisions with other particles (as described in Section 3.2.4); however in PIC, movement is due to electromagnetic forces acting on the ions and electrons in the plasma, which means the electromagnetic background presented in Section 4.1 is required to fully understand PIC simulation.

4.1.1 Maxwell's Equations and the Lorentz Force

The following list of equations are Maxwell's equations, a set of coupled partial differential equations that when combined with the Lorentz Force Law (defined in Equation 4.5) form the basis for electrodynamics.

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \tag{4.1}$$

$$\nabla \cdot \vec{B} = 0 \tag{4.2}$$

$$\nabla \times \vec{E} = \frac{\partial \vec{B}}{\partial t} \tag{4.3}$$

$$\nabla \times \vec{B} = \mu_0(\vec{J} + \epsilon_0 \frac{\partial \vec{E}}{\partial t}) \tag{4.4}$$

In Maxwell's equations, $\vec{E}$ is the electric field, $\vec{B}$ is the magnetic field, ρ is the electric charge density, and $\vec{J}$ is the current density. The constants ϵ_0 (vacuum permittivity) and μ_0 (vacuum permeability) are also present throughout the equations. This equation is the Lorentz Force Law.

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \tag{4.5}$$

In the context of PIC simulation, Maxwell's equations are used to solve for the electric and magnetic fields based on the charge distribution and other imposed boundary conditions.

4.1.2 Penning Trap

A Penning trap is a device that uses a combination of electric and magnetic fields to trap charged particles (such as the electrons and ions found in an ionized plasma). In most Penning Traps (such as the TITAN Penning Trap shown in Figure 4.1), the ions are confined to a plane via the application of a strong $\vec{B}$ field, and then the size of that plane is limited by applying an $\vec{E}$ field.

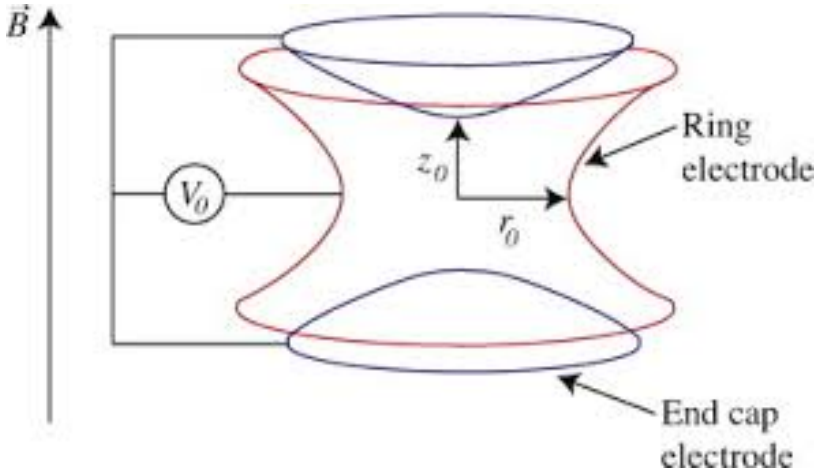


Figure 4.1: The TITAN Penning Trap currently in operation at TRIUMF in Vancouver, B.C. [24].

General Fusion uses a Penning Trap in the PI-3 ionization process (see Section 2.3 for more details). This Penning Trap is sketched in Figure 4.2, and uses a magnetic field to constrain electrons and ions to field lines with the high wall voltage further constraining electrons causing chain ionization to occur. This field lines for this process are shown in Figure 4.2. After the gas has been fully ionized, the Penning trap is turned off, and the ionized plasma moves towards the end of the chamber (towards the wider right side of Figure 4.2).

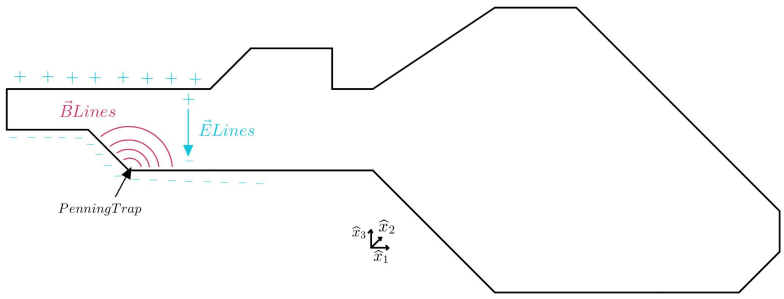


Figure 4.2: Sketch of General Fusion's Penning Trap during the ionization stage of plasma formation.

4.2 Plasma Physics Background

In this section, different plasma physics concepts that will be important in understanding PIC simulations are introduced. Readers that have experience working with plasmas and ionization can probably continue on to the next section.

4.2.1 Debye Length

The Debye length (typically represented by λ_D) is a measure of how far the electrostatic effect of a charge carrier persists in a plasma. In a plasma, charges are screened very quickly. For example, if a wall is positively charged, it will attract electrons and repel ions, leading to a smoothing of its potential gradient (as can be seen in Figure 4.3). This behavior follows the figure on the right, because if the electric field doesn't at least partially penetrate the plasma, then it is impossible for electrons and ions to actually compensate for the electric field and screen it out in the first place.

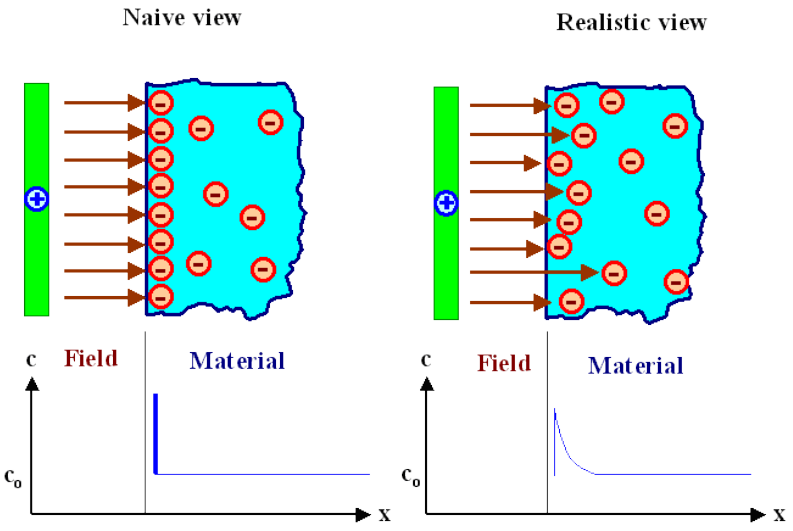


Figure 4.3: Comparison of physically impossible 'full shielding' (left) and Debye shielding (right) [25].

The Debye length decreases by a factor of $1/e$ with each Debye length λ_D from the charge source, where the unitless quantity λ_D can be represented by Equation 4.6 for a non-isothermal plasma with a background medium that may be treated as a vacuum.

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B / q_e^2}{n_e/T_e + \sum_j z_j^2 n_j / T_j}} \tag{4.6}$$

Where λ_D is the Debye length, ϵ_0 is the permittivity of free space, k_B is the Boltzmann constant, q_e is the charge of an electron, T_e and T_i are the temperatures of the electrons and ions, respectively, n_e is the density of electrons, n_j is the density of atomic species j, with positive ionic charge $z_j q_e$. However, the plasmas that will be studied in PI-3 ionization processes have very small ion mobility die to the small ionization timescale. This means that Equation 4.6 can be simplified by dropping the ion term, as seen in Equation 4.7.

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{n_e q_e^2}} \tag{4.7}$$

The Debye length is important for PIC simulations because if there is an imposed electric field boundary condition, then the cell size must be smaller than the Debye length to properly accommodate this boundary condition, otherwise the solution to the system of equations stated in 4.3 is unstable.

4.2.2 Quasi-Neutrality

Plasma systems that are formed by ionizing gas are typically able to be described as quasi-neutral. Quasi-neutral plasmas are effectively neutrally charged in small volume elements dV of the plasma despite being made up exclusively of charged particles (in the case of a fully ionized plasma). This is because the neutral gas that makes up the plasma is the source of electrons to the plasma, so the charges are balanced as in Equation 4.8.

$$Q_{ion} = Q_{electrons} = n_{valance} \cdot q_e \tag{4.8}$$

The condition of quasi-neutrality is important in understanding the properties of a specific plasma, and how the electromagnetic vector fields acting on the plasma impact its dynamics. For a plasma to be quasi-neutral, the Debye length (see Section 4.2.1 for more details about the Debye length) must be much smaller than the physical size of the plasma; this implies that the boundary effects of the plasma do not impact the internal bulk interactions of the plasma, and that these bulk interactions are more important than the edge interactions of the plasma.

4.2.3 Interaction Cross Sections

Microscopic interaction cross sections are used to determine the likelihood of two particles interacting. The true likelihood of such events is calculated or experimentally determined using quantum mechanics, but cross sections provide us with a classical analogue for this by giving an effective cross sectional area of a two particle system (made up of p_1 and p_2 as it moves through space with relative velocity $v_{rel} = v_{p_2} - v_{p_1}$. There are two main categories of interaction cross sections for plasmas.

1. Momentum and energy transfer
2. Plasma creation and destruction

Momentum and Energy Transfer

Table 4.1: Table containing information about momentum and energy transfer cross sections.

Momentum and Energy Transfer Collision Cross Sections		
Name	**Particles**	**Cross Section Symbol**
Neutral-Neutral	*neutral atom*, *neutral atom*	σ_{nn}
Neutral-Electron	*neutral atom*, e^-	σ_{ne}
Neutral-Ion	*neutral atom*, *charged ion*	σ_{ni}
Electron-Electron	e^-, e^-	σ_{ee}
Electron-Ion	e^-, *charged ion*	σ_{ei}
Ion-Ion	*charged ion*, *charged ion*	σ_{ii}

In most fusion plasmas, it is typical for energy to be of the same order of magnitude across different components of the plasma ($E_{ion} \approx E_{electron}$), which means momentum varies drastically between electrons and ions due to their mass discrepancy. This is due to electrons usually being the primary heat conductors in the plasma due to their higher mobility relative to heavier ions.

Plasma Creation and Destruction

In this thesis (and most fusion devices) the dominant plasma creation mechanism is electron impact ionization, and the dominant plasma destruction mechanism is 3-body recombination [26].

Table 4.2: Table containing information about momentum and plasma creation and destruction cross sections.

Plasma Creation and Destruction Cross Sections		
Name	**Particles**	**Cross Section Symbol**

Electron Impact Ionization	e^-, *neutral atom*	$\sigma_{e^-,ionization}$
3-Body Recombination	*ion*, e^-, e^-	$\sigma_{3-b,recomb}$
Radiative Recombination	*neutral atom*, e^-, γ	$\sigma_{rad,recomb}$

Of the two recombination methods listed in Table 4.2, 3-Body Recombination accounts for 95% of plasma recombination, while Radiative Recombination accounts for the other 5% at typical fusion plasma conditions [26].

4.3 Particle In Cell (PIC)

The Particle in Cell method refers to a specific method of solving a specific class of Partial Differential Equations (PDEs). In PIC, individual particles are tracked in continuous phase space (Lagrangian Frame), and macroscopic quantities such as densities, currents, and pressures are calculated simultaneously on a stationary mesh (Eulerian Frame) using these individual particles.

4.3.1 Types of PIC Applications

Since PIC codes have both Lagrangian and Eulerian data available (as described in Section 4.3), there are three different ways to build the forcing functions that move charged particles in the simulation.

1. PM: Particle-Mesh
2. PP: Particle-Particle
3. P^3M: Particle-Particle and Particle Mesh

Particle-Mesh (PM)

In PM simulations, the forcing functions that move the charged particles in the simulation are only based on the macroscopic cell averaged field quantities of the simulation. They are often interpolated from surrounding cells to the location of the particle that is being moved to get a better localized idea of the values of the vector fields that are influencing the particle's motion.

Particle-Particle (PP)

In PP simulations, the forcing functions that move the charged particles in the simulation are only based on the other particles themselves. Often, the particles included for these force calculations are from the cell that the particle is contained in, along with directly adjacent cells. These parameters

are easy to calculate, but can lead to large statistical fluctuations if the sample of particles is too small.

Particle-Particle and Particle-Mesh (PP-PM or P^{3}M)

In P^{3}M calculations, the forcing functions that move the charged particles in the simulation are based on the macroscopic cell averaged field quantities and the other particles themselves. The relative weighting of these 2 contributing factors depends on the application.

4.3.2 Superparticles in PIC Simulations

In the same way that large numbers of particles are represented by equivalent particles or superparticles in DSMC, they are also represented by superparticles in PIC simulations. This has the same purpose of saving massive amounts of computational resources, but has an additional complication. In DSMC simulations, the only factor to keep track of with these large superparticles was the number of real particles that they represented; but with PIC superparticles the charge and the mass of the particles that they are representing must also be recorded. This is because calculations of field quantities depends on the total charge and its distribution, so to properly calculate the $\vec{E}$ and $\vec{B}$ fields, total charge of the superparticle is required. Additionally, the charge to mass ratio of a superparticle must be the same as each of the particles that it represents, because the charge to mass ratio is what determines a particle's path through $\vec{E}$ and $\vec{B}$ fields.

4.3.3 Key Assumptions

The following five assumptions are made about the characteristics of the PIC simulation with regards to the number of particles, physical parameters, and the mesh geometry. These assumptions must be true to lead to physically relevant results from the PIC simulation. They are the same as the assumptions for DSMC with one additional assumption due to the role that $\vec{E}$ and $\vec{B}$ fields play.

1. Molecules move without interaction in free flight for the length of the chosen time step [15].
2. Impact parameters and initial orientations of colliding particles are random [15].
3. In every cubic mean free path there are many molecules, but only a small fraction need to be simulated for the molecular description of the flow to be accurate [15].
4. The only molecules that any molecule could collide with are the molecules within a molecule's finishing cell.
5. The grid spacing is small enough to resolve small variations in vector fields acting on the Lagrangian superparticles, and the solution to Maxwell's Equations is stable.

Only key assumption 5. will be further discussed here, as the others were described in detail in Section 3.2.1.

Grid Spacing

For resolution of small variations in vector fields, especially for imposed boundary conditions, the grid size for a PIC simulation must be smaller than the Debye length of the plasma. This can be calculated using Equation 4.6 or Equation 4.7 depending on the case.

4.3.4 PIC Algorithm

The PIC algorithm is very similar to the DSMC algorithm outlined in Section 3.2.4, and it is visualized in Figure 4.4.

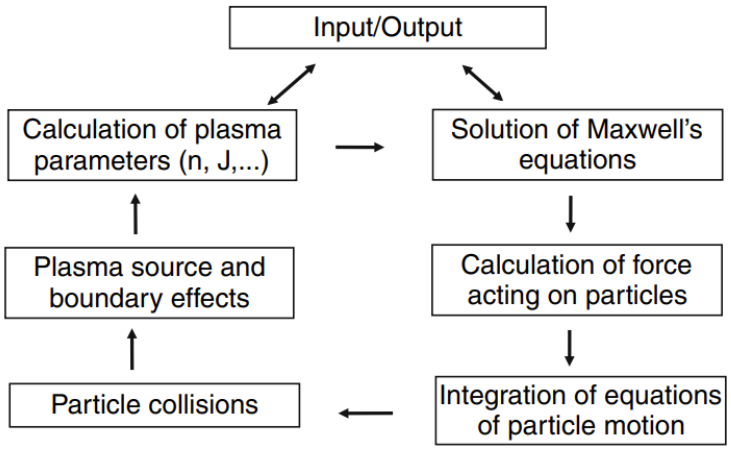


Figure 4.4: Scheme of a PIC simulation [27].

These steps of the PIC algorithm can also be represented as a list of tasks to be completed. Unlike DSMC however, some of these tasks are interrelated, so the process is less linear than in the DSMC case.

1. Calculation of plasma parameters (n, J, ...)
2. Solve Maxwell's equations
3. Calculate forces acting on particles
4. integrate equations of motion for each particle
5. Perform stochastic collisions between particles
6. incorporate boundary and source conditions

Calculation of Plasma Parameters

For most PIC simulations of all three types (PM, PP, P^3M), macroscopic plasma quantities are calculated at each time step (or at least at every time step where data is written to log files). However, for PM and P^3M PIC simulations these plasma parameters feed into the calculations of forces acting on particles (described in Section 4.3.4), so this is done at the beginning of each time step with the exception of the first time step, as the generation of particles to fit the specified initial conditions proceeds this. The choice of which parameters are calculated, and the methods of calculation for those parameters differ in different PIC packages. Usually the number densities of ions, electrons and neutrals (n_i, n_e, and n_n respectively) are calculated along with the net charge currents $J_{net} = J_i + J_e$ so that these can be used in the solving of Maxwell's equations.

Solve Maxwell's Equations

Maxwell's equations themselves are outlined in Section 4.1.1, and the study of the efficiency of different numerical methods in solving Maxwell's equations is a field of study developing independently of plasma physics and PIC simulation. More details about one such method are available in this paper [28], but in this thesis, we will treat it as if it is a solved problem. The PIC simulation needs to solve Maxwell's Equations to construct the localized $\vec{E}$ and $\vec{B}$ fields so that particle trajectories can be integrated properly to include the impact of electromagnetic forces.

Calculate Forces Acting on Particles

Once the $\vec{E}$ and $\vec{B}$ fields have been determined by solving Maxwell's equations, the forces acting on the particles can also be determined. These two fields exert a combined force defined as the Lorentz force (shown in Equation 4.5). This force is trivial to apply to a charged particle once the vector fields $\vec{E}$ and $\vec{B}$ have been determined. This must be done locally though, so usually the vector quantities are linearly interpolated to get localized values at each particle's position rather than just using the value at the cell's centre.

Integrate Equations of Motion for Each Particle

In DSMC simulation, particles follow straight paths extrapolated from their current velocity (as described in Section 3.2), but in PIC simulations, the electromagnetic fields can cause the particles to follow curves instead. This requires equations of motion to be constructed from the force balance acting on a particle with the ability for those forces to evolve in time. These forces then lead to particle trajectories by integrating the equations of motion.

Perform Stochastic Collisions

In PIC, stochastic collisions are computed similarly to in DSMC. Just like DSMC, we can visualize a 3-D volume of space that the particle has interacted with during the time step, and then determine whether or not it has interacted with any particles during that time step using Equation 4.9. This equation needs one alteration in the form of a representative cross section (like those described in Section 4.2.3).

$$P_{PIC} = \sigma_{interaction} W_p v_{rel} \Delta t_{PIC} / V_{PIC} \tag{4.9}$$

This calculation must be performed for each possible collision with another species for each individual particle. For example, if we are considering the movement of an ion, we would need to consider σ_{ii}, σ_{ie}, σ_{in}, and $\sigma_{3-b,recomb}$ from Section 4.2.3, with each depending on the presence of whichever species is interacting with the ion.

Incorporate Boundary Conditions and Plasma Sources

Finally, boundary conditions and potential internal sources are imposed on the code before macroscopic quantities are calculated headed into the next round of simulation. These boundary conditions could be imposed $\vec{E}$ and $\vec{B}$ fields at edges, or an influx of particles to the simulation domain.

4.4 Existing Open Source PIC Packages

This section outlines various modern PIC packages that have been designed for creating numerical solutions to different plasma equilibrium problems. They are listed in chronological order of first release, and the first line of each entry shows that release date, along with the most recent update, where *present* means that there are more features continually being added.

4.4.1 hyStrath

2014-present [20]**:** *hyStrath Website*

The GitHub repository *hyStrath* is an OpenFOAM extension that features hypersonic and rarefied gas dynamics code [20]. The code is specifically designed for atmospheric re-entry analysis, but is suitable for all kinds of rarefied gas dynamics problems, and has a module called pdFoam which is a hybrid electrostatic PIC and DSMC solver. It is also built within the OpenFOAM framework and strictly adheres to OpenFOAM's coding practices, so it is easy to expand their existing code base to add additional features.

4.4.2 Smilei PIC

2016-present [29]**:** *Smilei Website*

Smilei is a collaborative open source particle in cell code for plasma simulation [29]. Smilei was developed by HPC specialists and physicists for high performance on parallel supercomputers. It also has a built in python UI, and the ability to impose boundary conditions that are specified by python functions. It doesn't interface with any existing post processing tools, but does have some built in post-processing functions.

4.4.3 Starfish

2016-present [22]**:** *Starfish Website*

Starfish is a 2-D gas and plasma simulation code [22] developed by Particle In Cell Consulting (PIC-C). For modelling plasma, it uses the Electrostatic Particle in Cell method; and for modelling free molecular flow it can use several continuum solvers as well as DSMC. There are two versions. One is free and open source but contains limited features, and the other is only internally available within PIC-C, but can be requested in a compiled format.

4.4.4 picFOAM

2020-present [30]**:** *picFOAM Website*

The GitHub repository *picFOAM* is an electrostatic particle in cell solver built for simulating equilibrium and non-equilibrium plasmas with complex collective behaviour between different charged species [31]. It was also built to understand the interaction between plasmas and external circuitry attached to the boundaries of the plasma, which is how the high voltage arcs are created in the PI-3 plasma injector.

4.4.5 OSIRIS

2022-present [32]**:** *OSIRIS Website*

OSIRIS is an electro-magnetic PIC code designed to run on both high performance computing environments and smaller less resource intensive setups [32]. This software provides many different pieces of additional functionality, with the most interesting for PI-3 ionization simulations being electron impact ionization and advanced initialization routines for setting up particle densities to correspond to initial $\vec{E}$ fields.

4.5 Comparison of Existing Open Source PIC Codes

The PIC codes from section will be evaluated in a simple comparison matrix shown in Table 4.3. I have run a number of test cases in all five of these PIC packages. I will be returning to General Fusion full time this May, and will likely have the opportunity to continue this project at some point. I thought it would be more valuable to gain more information about what is going to be the perfect tool for the job than to forcibly create another simulation without verifying its results. I will discuss these packages with my supervisor this summer and we can decide which one to proceed with.

Table 4.3: Comparison matrix for evaluation of different Open Source PIC codes.

Feature	hyStrath	Smilei	Starfish	picFOAM	OSIRIS
2-D	Yes	Yes	Yes	Yes	Yes
3-D	Yes	Yes	No	Yes	Yes
Axisymmetric	Yes	Yes	Yes	Yes	Yes
Built in Ionization	Yes	No	Yes	Yes	Yes
Mesh Generation Utility	Yes	Yes	No	Yes	Yes
$\vec{E}$ Field Boundary Conditions	Yes	Yes	Yes	Yes	Yes
Imposed $\vec{B}$ Field in Particle Motion	No	Yes	No	No	Yes
Documentation /10	6	8	6	4	7
Supporting Literature /10	7	7	6	8	5

Chapter 5

Neutral Gas Experimental Approach

The experimental approach for determining the neutral gas distribution at the moment of ionization is briefly summarized by the five steps below.

1. Clearly define the problem that is being addressed
2. Perform preliminary calculations to help narrow down what tools are viable for solving this problem
3. Explore different problem solving approaches
4. Validate the chosen approach and tools using test cases
5. Run the final simulations to get a solution

Throughout this chapter, the background from Chapters 2 and 3 will be referenced to provide additional context to experimental decisions and the problem definition.

5.1 Problem Definition

This thesis is jointly supervised by General Fusion and UTIAS, but the problem itself was brought forward by General Fusion from a series of conversations that I had with their head of Plasma Reconstruction Ryan Zindler. This problem definition was left very broad, which is a nice starting point to work from since it leaves open any available option. The problem definition given by General Fusion is quoted below.

> The background here is that we form our plasmas in a "Marshall Gun". Starting at high vacuum, we puff in a small amount of neutral (Deuterium) gas into the gun. We then fire a bank of capacitors, which puts a high voltage (10kV) across the gun, which then ionizes

> the gas, and magnetically accelerates it into the pot. This is called "formation", and we have code dedicated to simulating the formation process. However, what we don't yet have is a good way to predict the initial plasma state that we feed into our formation simulations.
>
> So part 1 of the project is to start from the gas puff nozzle and valve geometry, and deduce a density distribution of the neutral gas as a function of time and space.
>
> Part 2 would be to take the gas distribution from part 1, and model the breakdown dynamics to predict where the current will flow once the gas starts getting ionized. Read up on the derivation of the Paschen curve to understand this task. You won't be able to just use the Paschen curve, because magnetic fields change the results significantly.
>
> If you wanted to, you could reduce the scope by just doing one of the 2 parts above.

From this detailed description we can condense our problem into a simple problem definition that can be used when we are looking into what simulation tools are appropriate for solving this problem.

> This thesis will simulate the neutral gas distribution over time in an approximated version of the PI-3 plasma injector geometry to allow for a full density time distribution of the entire geometry to be produced.

5.2 Preliminary Calculations and Verification

5.2.1 Flow Regime for Valve Exit and Unfilled Chamber

General Fusion are interested in macroscopic properties such as the density and velocity distributions in their reaction chamber, so the flow regime needs to be properly defined so problem solving approaches can be properly scoped. Knudsen Number regimes were used to assess this and parameterize the flow (background information describing the Knudsen Number is used for this is presented in Sections 3.1.1 and 3.1.2). From Equation 3.1, calculating the Knudsen Number requires λ (the mean free path), and L (the length scale, provided by General Fusion). The mean free path can be calculated using Equation 5.1.

$$\lambda = \frac{1}{\sigma \cdot n} = \frac{1}{\sqrt{2}\pi d^2 \cdot n} \tag{5.1}$$

Where in Equation 5.1, Equation 5.2 is substituted to obtain the second equality.

$$\sigma = \sqrt{2}\pi d^2 \tag{5.2}$$

General Fusion supplied internal pressure measurements from both their ionization chamber and

their filling pump. These could be converted to number densities using the ideal gas law in the form of Equation 5.3.

$$PV = NRT \Longrightarrow \frac{N}{V} = n = \frac{P}{RT} \tag{5.3}$$

General Fusion use pure diatomic Deuterium as their fuel, and provided pressure values which were easily converted to number densities, and by extension mean free path estimates (using Equation 5.1). The approximate Knudsen Numbers of the vacuum and valve exit are presented in Table 5.1.

5.2.2 Flow Regime for Chamber after Filling

The two extreme values presented in Section 5.2.1 don't provide a good overall picture of the flow, so another Knudsen Number was calculated. This Knudsen Number was for the gas in the chamber after filling, assuming an even distribution in just the top portion of the chamber. This will be referred to as the steady state case, since if it was run on a real container of that size, and then left alone for a long period of time this is the state that it would reach.

To calculate the steady state Knudsen Number requires both a volume estimate and an estimate of the number of particles that would be in that volume. The volume estimate is purely geometric, and from an approximated PI3 geometry that will be used in later simulations it can be calculated very quickly by assuming that the radii are averaged on the sloped sections (for example, in the second section from the left in Figure 5.1, $r_{inner}(section2) \approx \frac{r_{inner}(section1)+r_{inner}(section3)}{2}$). Calculating the volume from the dimensions in Figure 5.1 can be quickly completed using Equation 5.4.

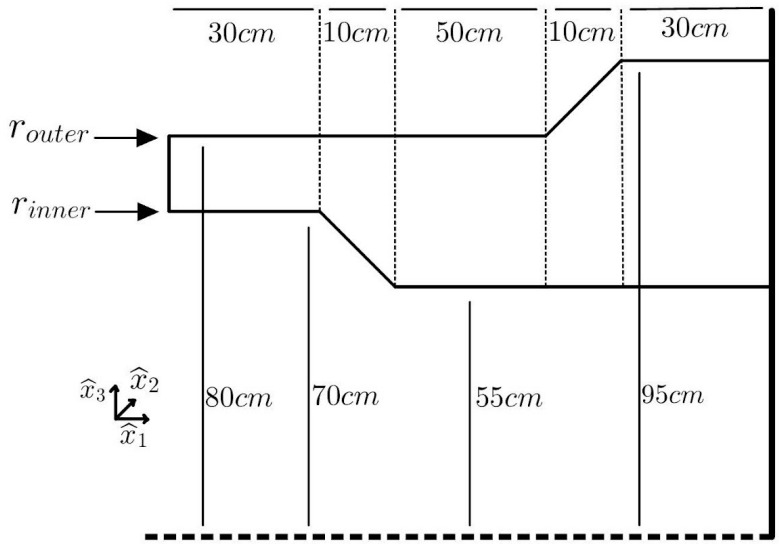


Figure 5.1: Sketch of approximated PI-3 geometry with reference dimensions for calculating an approximate volume. This drawing is axisymmetric about the dashed line at the bottom of the figure.

$$V_i = \pi(r^2_{outer,i} - r^2_{inner,i}) \cdot l_i \tag{5.4}$$

Where V_i, $r_{outer,i}$, $r_{inner,i}$, and l_i are the volume and length of the $i'th$ section. Then the total volume can be represented by Equation 5.5.

$$V = \sum_i V_i = \sum_i \pi(r^2_{outer,i} - r^2_{inner,i}) \cdot l_i \tag{5.5}$$

Calculating this volume gives $1.4m^3$.

Now the number of particles released into the chamber must be calculated. Since we know the number density of the input valve, if we assume that the number density of the input valve doesn't change during the length of the gas puff then we can calculate a constant flux and then use that to determine how many particles enter the chamber. This is mathematically shown in Equation **??**.

$$N_{particles} = \Gamma \cdot A_{valves} \cdot \Delta t = \frac{1}{4} n c_{avg} \cdot 25\pi r^2_{valve} \cdot \Delta t \tag{5.6}$$

In Equation 5.6, Γ is defined in Equation 3.3 and A_{valves} is the cross sectional area of all 25 input valves. $N_{particles}$ is now easily calculated since it is in terms of known quantities, and the value is $N_{particles} = 1.5 \cdot 10^{22}$ using $\Delta t = 600\mu s$. Δt was chosen to be $600\mu s$ because the current gas puff timescale used at General Fusion is $240\mu s$ so simulating past that gives a better idea of whether they should potentially be letting the gas puff run for longer. The number density from this number of particles and the approximated volume is presented in Table 5.1.

5.2.3 Table of Knudsen Numbers

Table 5.1: Mean Free Paths and Knudsen Numbers for General Fusion's Plasma Injector.

Chamber State	Number Density	Cross Section σ	Mean Free Path	Knudsen Number
Vacuum	$2 \cdot 10^{17}$	$8 \cdot 10^{-19}$ [33]	6	$3 \cdot 10^{1}$
Steady State	$1 \cdot 10^{22}$	$8 \cdot 10^{-19}$ [33]	$1 \cdot 10^{-4}$	$6 \cdot 10^{-4}$
Valve Exit	$2 \cdot 10^{26}$	$8 \cdot 10^{-19}$ [33]	$6 \cdot 10^{-9}$	$3 \cdot 10^{-8}$

These flows vary greatly from highly rarefied initially to effectively continuous as the gas puff is finishing. Modelling all of these modes simultaneously in one simulation will present a challenge with respect to finding a simulation software capable of completing it accurately using an acceptable amount of computational resources.

5.3 Approaches

Based on the problem definition from Section 5.1, the main functionality that we are looking for when deciding what type of simulation to run is a simulation type that can handle the full Knudsen number range of interest shown in Table 5.1.

5.3.1 List of Potential Approaches

The classes of simulations that could potentially address the problem as defined are listed below.

1. Moment closure
2. Fokker-Planck solver
3. Fokker-Planck and DSMC hybrid solver
4. Direct Simulation Monte Carlo (DSMC)
5. DSMC and Navier Stokes combined hybrid solver

5.3.2 Moment Closure

Moment closure is a method of solving for the moments of a stochastic process. The application of moment closure techniques to the Boltzmann equation of rarefied gas dynamics is a very interesting tool for this problem. If properly implemented it should be able to handle the entire flow regime in question for this problem as defined in Section 5.1. Unfortunately there were not any modern open source moment closure software packages that I could find with sufficient documentation for me to be confident using them as a tool for this thesis project; and the creation or even substantial modification of a moment closure software package is its own undergraduate thesis. For those reasons, moment closure will not be used in this thesis, but is something that will be shared with General Fusion as another approach to solving this problem.

5.3.3 Fokker-Planck Solver

The Fokker-Planck equation is a PDE that describes the time evolution of the probability density function of a particle under the influence of drag forces and other random forces (such as random collisions). Solutions to this equation can be imposed on Lagrangian type simulated particles like those in a DSMC simulation, but their accuracy decreases as $Kn \to 1$. This means that a Fokker-Planck solver on its own is not an appropriate tool for this problem.

5.3.4 Fokker-Planck and DSMC Hybrid Solver

If it was within the scope of this thesis to make a solver from scratch to perfectly handle this problem, an adaptive mesh refinement Fokker-Planck and DSMC hybrid solver like the Eiger rarefied gas dynamics solver constructed by Küchlin and Jenny at ETH Zurich [34]. Unfortunately that is outside the scope of an undergraduate thesis. There are no other such solvers that I could find which were open source, and the documentation surrounding their code is extremely limited so I was not comfortable using it in this thesis.

5.3.5 Direct Simulation Monte Carlo

DSMC is extensively described in Chapter 3, so its mechanics will not be outlined here. DSMC is an interesting tool because it *can* be used to simulate any gas dynamics problem and will always provide an accurate solution should the four key assumptions of DSMC presented in Section 3.2.1 be satisfied. The drawback to this flexibility and the relative simplicity of DSMC as a method is the computational expense. Running large DSMC simulations that are outside the transitional regime outlined in Figure 3.2 are incredibly expensive due to the high degree of collisionality requiring incredibly small time steps for the solution to remain accurate. With that being said, this is a great tool for tackling the large range of Knudsen Numbers that will be encountered in this simulation provided the time steps and mesh sizes are carefully chosen to accommodate this range. This capability along with the large number of well documented and widely verified DSMC packages is why DSMC was the chosen tool for this thesis.

5.3.6 Direct Simulation Monte Carlo and Navier Stokes Hybrid Solver

There are some available open source DSMC Navier Stokes hybrid solvers, and it was kept as an option in case the lower Knudsen number sections of the flow were causing the computational time to be too expensive, but it didn't end up being necessary to use one of these solvers as the DSMC code was able to run in an acceptable amount of time.

5.4 Test Cases

With the chosen approach of DSMC and the problem appropriately constrained, a specific DSMC package was chosen (see Section 3.4 for a comparison of available Open Source Packages). This package was hyStrath (see Section 3.3.3 for a brief overview), and information about a number of the other open source DSMC packages is available in Section 3.3. The code in hyStrath was developed by a collaboration of a number of universities, and has been extensively verified in terms of its core functionality [23].

5.4.1 Defining Specific Test Cases

A large portion of the reasoning behind these test cases is to verify that the code is being properly utilized, not to verify its core functionality. The following list of tests will be performed in order to make sure the results of the final simulation are valid. These four tests have been designed to be able to have intuitive visual solutions and numerically verifiable solutions. That is why the first three cases are functionally collisionless, so that they can be compared to simple collisionless models. Each test case is explained before its simulation results are presented in Chapter 6.

1. Low pressure filling from the bottom edge of a box
2. Low pressure filling from a slit in the bottom edge of a box
3. Low pressure filling in an approximated version of the PI-3 plasma injector
4. High pressure filling from the bottom edge of a box

Before any of the test cases were run, a quick verification framework of the key DSMC assumptions was constructed so the validity of these assumptions could be verified before taking the time and computational resources to run a simulation.

5.4.2 Verification Framework of DSMC Assumptions

Before running a new simulation, the four key DSMC assumptions are verified for the entire domain of the simulation. The key equations used for this are presented here, so they can be referenced later, and their results succinctly presented in tables following the Format of Table 5.2.

Assumption 1: Molecules move without interaction in free flight for the length of the chosen time step. This is quantified by Equation 5.7, where d is the average distance travelled in one time step (defined in Equation 5.8). Within Equation 5.8, c_{gas} is defined in Equation 3.4, and Δt is the length of one time step.

$$\lambda \gg d \tag{5.7}$$

$$d = c_{gas}\Delta t \tag{5.8}$$

Assumption 2: Impact parameters and initial orientations of colliding particles are random. This is a good assumption when Equation 5.9, where n_{max} is the highest possible number density, and $n_0 = 2.68684 \cdot 10^{25}$ is Lorschmidt's Number. More information about this is available in Section 6.1 of [15].

$$n_0 \gg n_{max} \tag{5.9}$$

Assumption 3: n every cubic mean free path there are many molecules, but only a small fraction need to be simulated for the molecular description of the flow to be accurate. This is true if Equation 5.10 holds within each cell. This means that there are greater than 20 representative DSMC particles in each cell once there are a substantial number of collisions occurring.

$$N_{DSMC} > 20 \tag{5.10}$$

Assumption 4: the only molecules that any molecule could collide with are the molecules within a molecule's finishing cell. This assumption holds when the particle travels less than the length of the side of the cell, which is shown mathematically in Equation 5.11.

$$L_{cell} \gg d \tag{5.11}$$

The table below is how the verification of DSMC assumptions will be presented for each of the following test simulations as well as the final simulation in Section 5.5.

Table 5.2: Example of DSMC assumptions verification table.

Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y/N	Y/N	Y/N	Y/N

5.5 Final Simulation

The final simulation is a culmination of all the different test simulations. The initial low pressure test case of a box with the whole bottom face being an inflow patch was to verify that the simulation package was being used correctly. The second case was to make sure outflow from a point was working correctly in a controlled case, so that the third case with novel geometry could be compared to that solution right around the inflow patch since the local dynamics should be similar. The Fourth simulation with a full side inflow patch at high pressure was to make sure it was going to be computationally feasible to complete a high pressure run with the full scale geometry. Finally the full simulation took all these concepts and integrated them to determine an accurate picture of the gas dynamics in General Fusion's PI-3 plasma injector.

Chapter 6

Neutral Gas Simulations

All neutral gas simulations in this section were run using the hyStrath dsmcFoam+ package. See Section 3.3.3 for a brief summary, Section 3.4 for some reasons why hyStrath was chosen, Section 5.3.1 for why DSMC was chosen, and Appendix B for links to a tutorial to download hyStrath.

The following table contains the key parameters for all five main simulations that were run to create the final PI-3 neutral gas distribution. This table combined with the geometry files which can be downloaded from the *GitHub repository* (more information about the GitHub repository is available in Appendix A) associated with this thesis can be used to recreate all the results that are presented in this chapter.

Table 6.1: DSMC simulations settings for all 5 primary simulations.

Simulation	1 - Section 6.1	2 - Section 6.2	3 - Section 6.3	4 - Section 6.4	5 - Section 6.5
n_{inlet}	$2.5 \cdot 10^{18}[1/m^3]$	$2.5 \cdot 10^{18}[1/m^3]$	$2.5 \cdot 10^{18}[1/m^3]$	$1.5 \cdot 10^{26}[1/m^3]$	$1.5 \cdot 10^{26}[1/m^3]$
n_{vacuum}	$0[1/m^3]$	$0[1/m^3]$	$0[1/m^3]$	$1 \cdot 10^{17}[1/m^3]$	$1 \cdot 10^{17}[1/m^3]$
$n_{equivalent}$	$2 \cdot 10^{10}[1/m^3]$	$1 \cdot 10^{9}[1/m^3]$	$1 \cdot 10^{8}[1/m^3]$	$1 \cdot 10^{18}[1/m^3]$	$1 \cdot 10^{13}[1/m^3]$
T_{inlet}	$300[K]$	$300[K]$	$300[K]$	$300[K]$	$300[K]$
d_{cell}	$10[mm]$	$15[mm]$	$5[mm]$	$10[mm]$	$2.5[mm]$
$\Delta t_{timstep}$	$100[ns]$	$3[\mu s]$	$300[ns]$	$100[ns]$	$100[ns]$
$t_{simulation}$	$600[\mu s]$	$600[\mu s]$	$600[\mu s]$	$600[\mu s]$	$600[\mu s]$

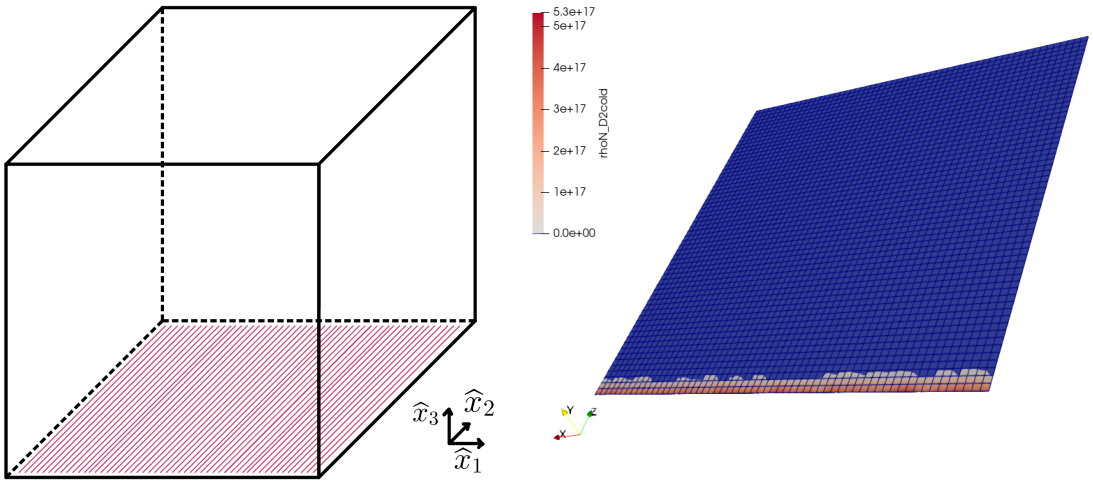
6.1 Low Pressure Filling From the Bottom Edge in a Box

The first simulation that was run was a collisionless filling simulation within a rectangular prism domain. The entire bottom face of the box is an inlet slot as shown in Figure 6.1a, and this case

was constructed so that the numerical validity of a nearly collisionless solution could be tested to ensure that the code was being properly utilized.

6.1.1 Geometry and Simulation Mesh

The geometry of this is a cube that was compressed down to effectively 2 dimensions by shrinking the size of $\hat{x_2}$ seen in Figure 6.1a drastically. This shrinking of $\hat{x_2}$ forms the mesh seen in Figure 6.1b. The overall dimensions of this rectangular prism are $(1m, 10^{-3}m, 1m)$.



(a) Sketch of the simulation geometry

(b) Mesh that was generated for the simulation.

Figure 6.1: Diagrams of the simulation geometry used in Section 6.1, where the darker red part of the mesh is right above the inlet.

6.1.2 DSMC Assumption Verification

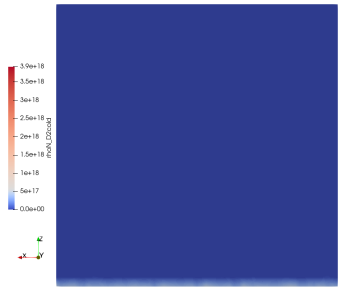
After constructing an appropriate mesh, the assumptions presented in Section 5.4.2 were calculated to see if they held, and the results of those calculations are in Table 6.2.

Table 6.2: DSMC assumptions verification table for low pressure filling from a bottom edge slit.

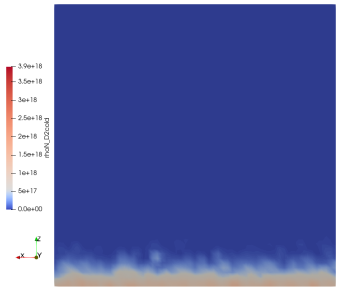
Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y	Y	Y	Y

6.1.3 Time Evolution of Density in the Entire Domain

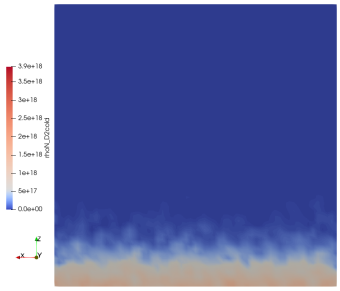
The main feature for this case was seeing a front of higher density gas filling in from the bottom of the chamber. The time-density plot at a localized point on the front is discussed in Section 6.1.4. From an eye test perspective, the plots in Figure 6.2 look very promising.



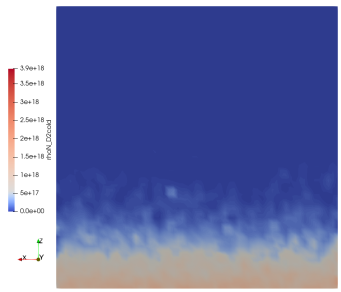
(a) Density distribution at $t = 15\mu s$.



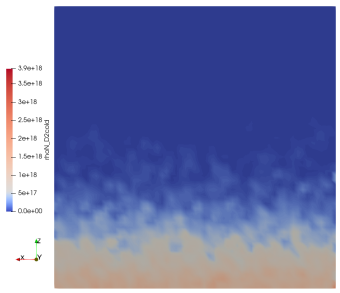
(b) Density distribution at $t = 75\mu s$.



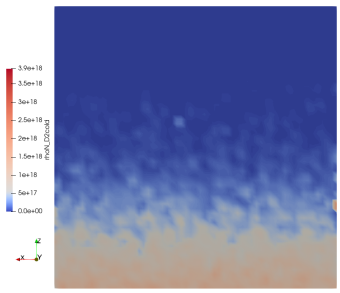
(c) Density distribution at $t = 135\mu s$.



(d) Density distribution at $t = 195\mu s$.



(e) Density distribution at $t = 255\mu s$.



(f) Density distribution at $t = 315\mu s$.

Figure 6.2: Plots of the time evolution of the density distribution for a gas filling process in a cube with an inlet patch as the bottom edge. (high resolution animation available on the GitHub from Appendix A)

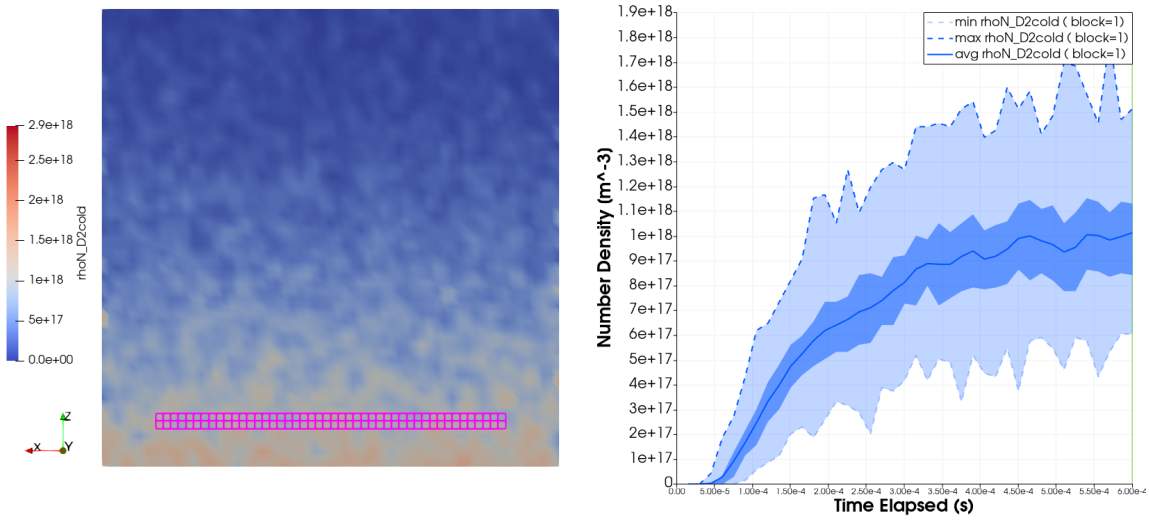
6.1.4 Verification of Simulation Results

The plot in Figure 6.3b is a density plot over the duration of the simulation, so the blue line is the mean density in the selected cells shown in Figure 6.3a. The medium blue surrounding it is the separating line separating out a quartile (meaning 50% of the densities lie within this range, with 25% on either side of the blue line) and the light blue outside region contains the outer 50% of values.

From this plot, we expect to see an initial spike at around $100\mu s$. This is because the case is largely collisionless, and for a case of collisionless filling we can approximate the inflow as a Boltzmann distribution provided we aren't interested in being as precise as to account for the velocity bias of the leading edge of particles. We can use Equation 6.1 to calculate the most commonly occurring velocity in the Boltzmann distribution ($\hat{c}$).

$$\hat{c} = (\frac{2kT}{m})^{\frac{1}{2}} \tag{6.1}$$

Then from that value of $\hat{c}$, we can get an approximation of when we would expect to see the first real density spike using the fact that our cells of interest are centred around a distance of $10cm$ from the



(a) Simulation plot showing cells included in the plot of Figure 6.3b.

(b) Plot of density over time in the low pressure bottom edge filling simulation, $10cm$ from the inlet.

Figure 6.3: Plot of density over time, with the cells selected shown on the left in Figure 6.3a and the plot itself shown on the right in Figure 6.3b.

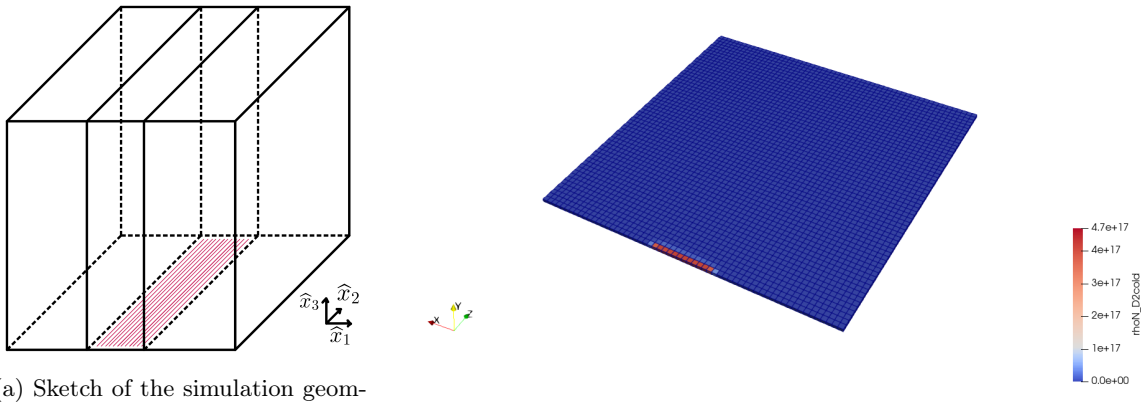
inflow. This gives us $t = \frac{d}{v} \approx 100\mu s$. Just before this time we see the first spike, and at $100\mu s$ the density is at 20% of the final plateau that is reached before back wall reflection can be accounted for. This means that the code is being properly utilized for collisionless flows.

6.2 Low Pressure Filling From the Bottom Edge Slit in a Box

After the first simulation confirmed that the code was being properly utilized, the goal of the second simulation was to make sure that divergence from a point was working correctly as well, since this is a key component of the final simulation in Section 6.5. It will be tested on a simpler case of a box with an inlet slit to verify that the divergence is uniform, and then a low density test case of an approximated PI-3 plasma injector geometry will be run.

6.2.1 Geometry and Simulation Mesh

The geometry of this simulation is very similar to that of Section 6.1, in that it is a cube that was compressed down to effectively 2 dimensions by shrinking the size of $\hat{x_2}$ seen in Figure 6.4a drastically. This shrinking of $\hat{x_2}$ forms the mesh seen in Figure 6.4b. The overall dimensions of this rectangular prism are $(1m, 10^{-3}m, 1m)$.



(a) Sketch of the simulation geometry

(b) Mesh that was generated for the simulation.

Figure 6.4: Diagrams of the simulation geometry used in Section 6.2, where the darker red part of the mesh is right above the inlet.

6.2.2 DSMC Assumption Verification

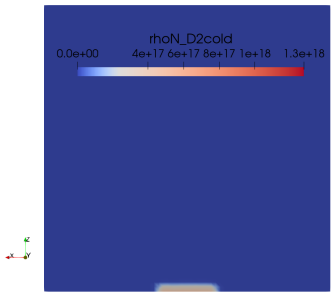
After constructing an appropriate mesh, the assumptions presented in Section 5.4.2 were calculated to see if they held, and the results of those calculations are in Table 6.3.

Table 6.3: DSMC assumptions verification table for low pressure filling from a bottom edge slit.

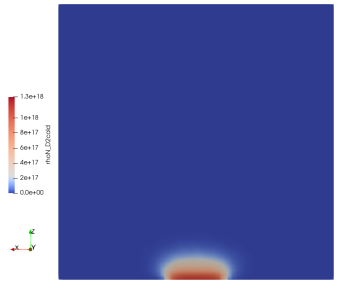
Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y	Y	Y	Y

6.2.3 Time Evolution of Density in the Entire Domain

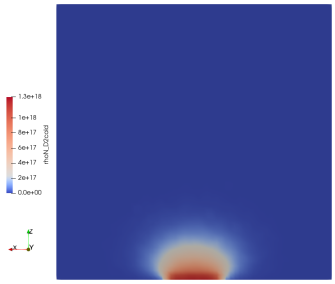
The main feature of interest for this case was uniform divergence from the infill slit as time progresses. We can see from the series of plots in Figure 6.5 that there is uniform divergence from the slit. In this simulation there was no need to recompute the values calculated in Section 6.1.4 since the collisionless Boltzmann assumption settings have not changed. The additional information gained from this simulation can be verified by looking at the time evolution of the density distribution throughout the domain, since that in itself shows the uniform divergence that this simulation was verifying.



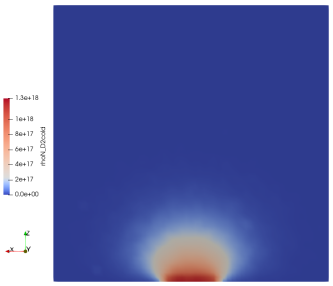
(a) Density distribution at $t = 15\mu s$.



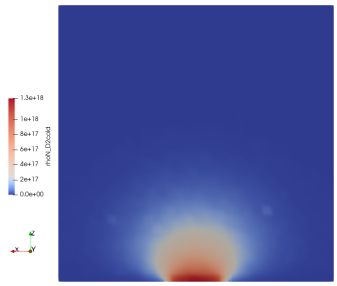
(b) Density distribution at $t = 75\mu s$.



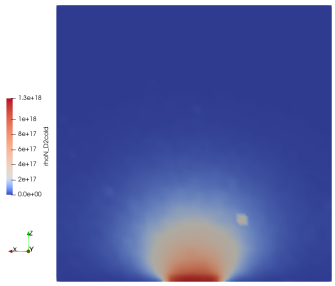
(c) Density distribution at $t = 135\mu s$.



(d) Density distribution at $t = 195\mu s$.



(e) Density distribution at $t = 255\mu s$.



(f) Density distribution at $t = 315\mu s$.

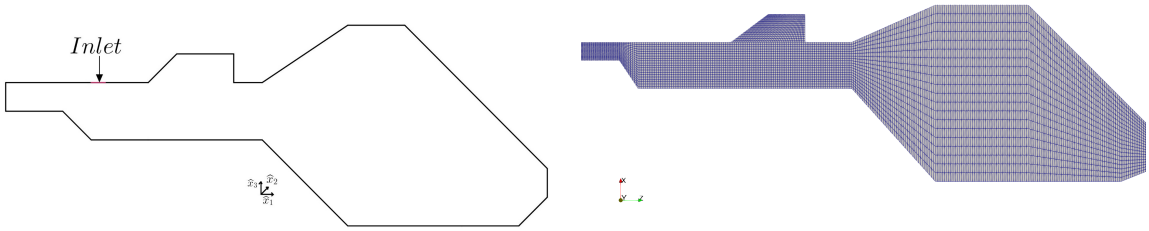
Figure 6.5: Plots of the time evolution of the density distribution for a gas filling process in a cube with an inlet slit. (high resolution animation available on the GitHub from Appendix A)

6.3 Low Pressure Filling in an Approximated PI-3 Geometry

After using the first two simulations (described in Sections 6.1 and 6.2) to show that the software was being properly utilized for known geometries, it was time to create an approximated mesh for General Fusion's PI-3 plasma injector. This was a major task on its own, as there were 54 vertices and 12 different blocks that made up the mesh which all had to be specified in a geometry file (which is available in the *GitHub repository*).

6.3.1 Geometry and Simulation Mesh

There was no NDA associated with this project which meant work could commence right away, but also lead to some approximations being made. The geometry used in the PI-3 simulations is based on reading approximate vertices of a plot. The pot on the right side of the real PI-3 plasma injector is smoothly curved, not jagged as shown in Figure 6.11, but that part of the domain should not play a role in the dynamics of the neutral gas filling case. The features that play a role in that are representative enough that the results of the final simulation will provide relevant context for General Fusion.



(a) Sketch of the simulation geometry

(b) Mesh that was generated for the simulation.

Figure 6.6: Diagrams of the simulation geometry used in Section 6.3, where inlet is labeled on the sketch.

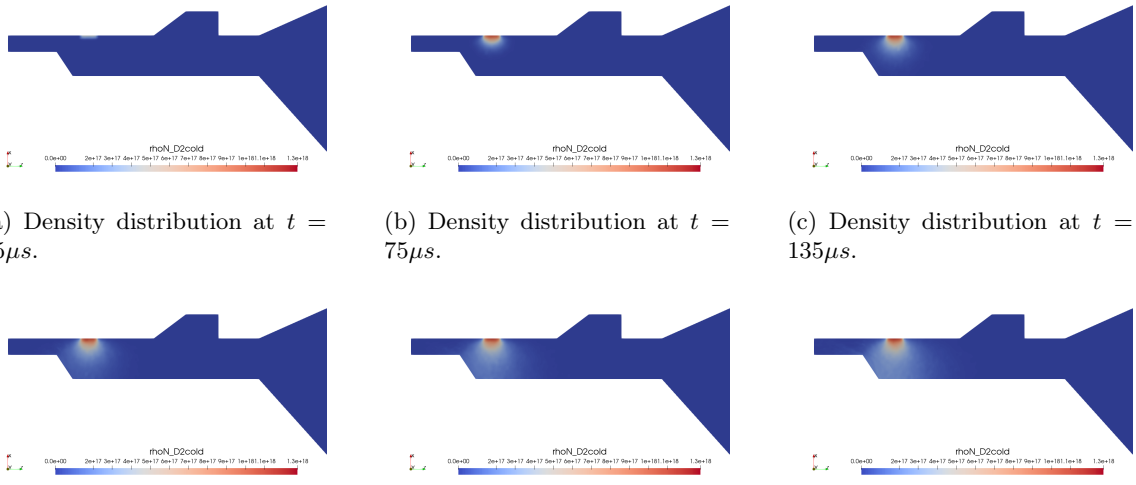
6.3.2 DSMC Assumption Verification

After constructing an appropriate mesh, the assumptions presented in Section 5.4.2 were calculated to see if they held, and the results of those calculations are in Table 6.4.

Table 6.4: DSMC assumptions verification table for low pressure filling in an approximated PI-3 plasma injector.

Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y	Y	Y	Y

6.3.3 Time Evolution of Density in the Entire Domain



(a) Density distribution at $t = 15\mu s$.

(b) Density distribution at $t = 75\mu s$.

(c) Density distribution at $t = 135\mu s$.

(d) Density distribution at $t = 195\mu s$.

(e) Density distribution at $t = 255\mu s$.

(f) Density distribution at $t = 315\mu s$.

Figure 6.7: Plots of the time evolution of the density distribution for a gas filling process in an approximated PI-3 plasma injector at low inlet pressure. (high resolution animation available on the GitHub from Appendix A)

This simulation of the final Pi-3 geometry at low pressure was used to test that there weren't any strange phenomena caused by the interaction between the geometry and the implementation of hyStrath, since this mesh is more complex than the meshes in any of the test cases associated with hyStrath. It was helpful to run this at lower number densities, since the simulations ran faster due to the lower number density and they had less complex filling dynamics since they were collisionless. There ended up being about six versions of the geometry before all the meshes were working well together and all the different simulation blocks were functioning correctly. This was all verified by eye at this stage.

6.4 High Pressure Filling From the Bottom Edge of a Box

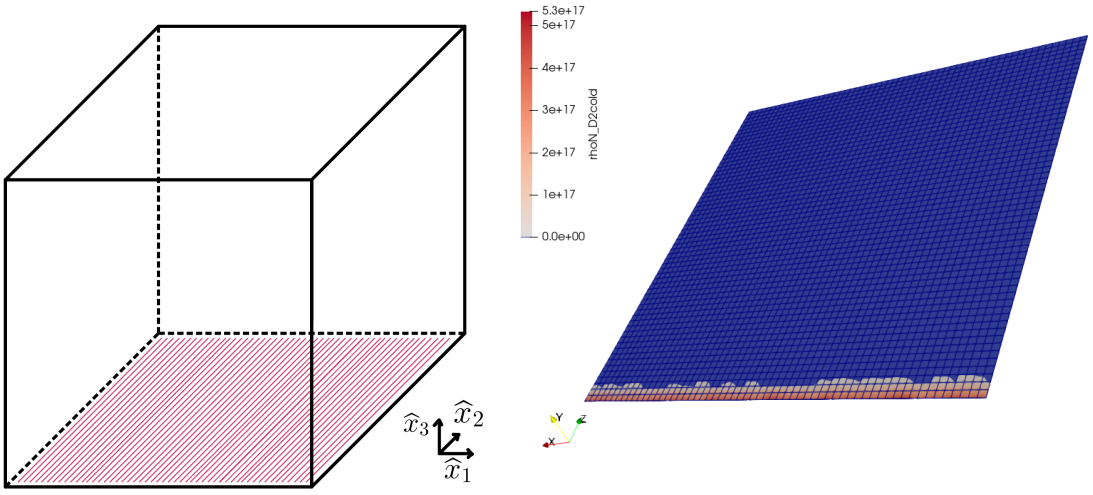
After the first three low density simulations from Sections 6.1, 6.2, and 6.3; the final stage of verification was to make sure that a final simulation would produce meaningful results at operational inflow pressures. Up to this point, low infill pressures were used so that the mean free path λ was always on the order of the length scale of the problem L, leading to a Knudsen Number of $Kn \approx 1$. With this specific setup, the collisionality of particles can be seen visually and in the simulation logs as the collision counter increments, but the cases can be treated as functionally collisionless when looking to verify their dynamics. This was a great strategy for testing, but a highly collisional case should be tested prior to running the final test case in Section 6.5, and using the same simple geometry as in Section 6.1 was the best way to have a verifiable case. This case was also used to verify that running a PI-3 simulation would be computationally feasible at this pressure, otherwise hybrid DSMC Navier Stokes solvers like those mentioned in Section 5.3.6 would have been explored.

6.4.1 Geometry and Simulation Mesh

This geometry is the same as the geometry in Section 6.1.1, but the mesh is finer. Calculations for collisions can slow down if there are too many particles in one cell, so the mesh was made finer to stay closer to 10^2 to 10^3 maximum particles per cell. The overall dimensions of this rectangular prism are $(1m, 10^{-3}m, 1m)$.

6.4.2 DSMC Assumption Verification

After constructing an appropriate mesh, the assumptions presented in Section 5.4.2 were calculated to see if they held, and the results of those calculations are in Table 6.5.



(a) Sketch of the simulation geometry

(b) Mesh that is generated for the simulation.

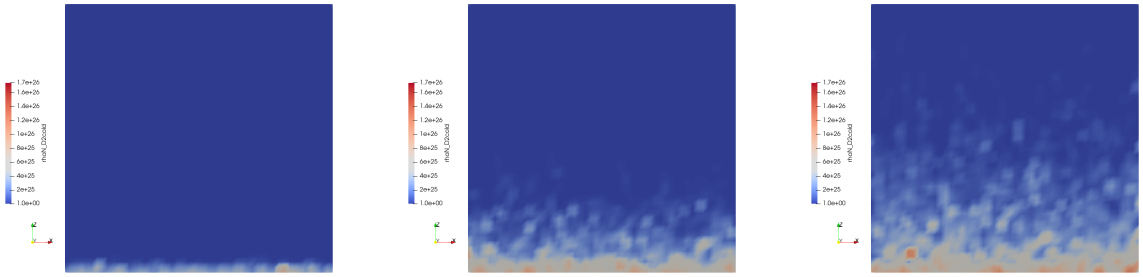
Figure 6.8: Diagrams of the simulation geometry used in Section 6.4, where the darker red part of the mesh is right above the inlet.

Table 6.5: DSMC assumptions verification table for neutral gas filling of an approximated PI-3 plasma injector at operational inlet pressure.

Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y	Y	Y	Y

6.4.3 Time Evolution of Density in the Entire Domain

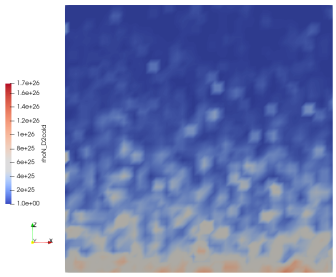
This simulation ended up having a larger number of DSMC superparticles than were predicted for the final PI-3 geometry due to the difference in inflow patch size. That meant that the number of equivalent real particles that needed to be represented by each superparticle was higher, and that lead to a lot of noise despite this simulation being fairly computationally expensive to run. This noise shouldn't greatly impede its ability to be used as a verification test for higher inlet pressure flows. Far lower noise was present in the final simulation in Section 6.5.



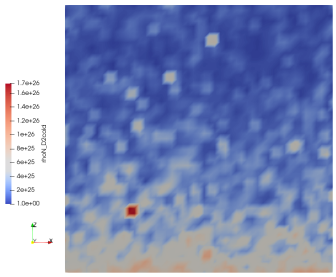
(a) Density distribution at $t = 15\mu s$.

(b) Density distribution at $t = 75\mu s$.

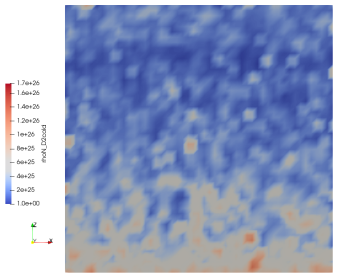
(c) Density distribution at $t = 135\mu s$.



(d) Density distribution at $t = 195\mu s$.



(e) Density distribution at $t = 255\mu s$.



(f) Density distribution at $t = 315\mu s$.

Figure 6.9: Plots of the time evolution of the density distribution for a gas filling process in a cube with the bottom edge as an inflow patch. (high resolution animation available on the GitHub from Appendix A)

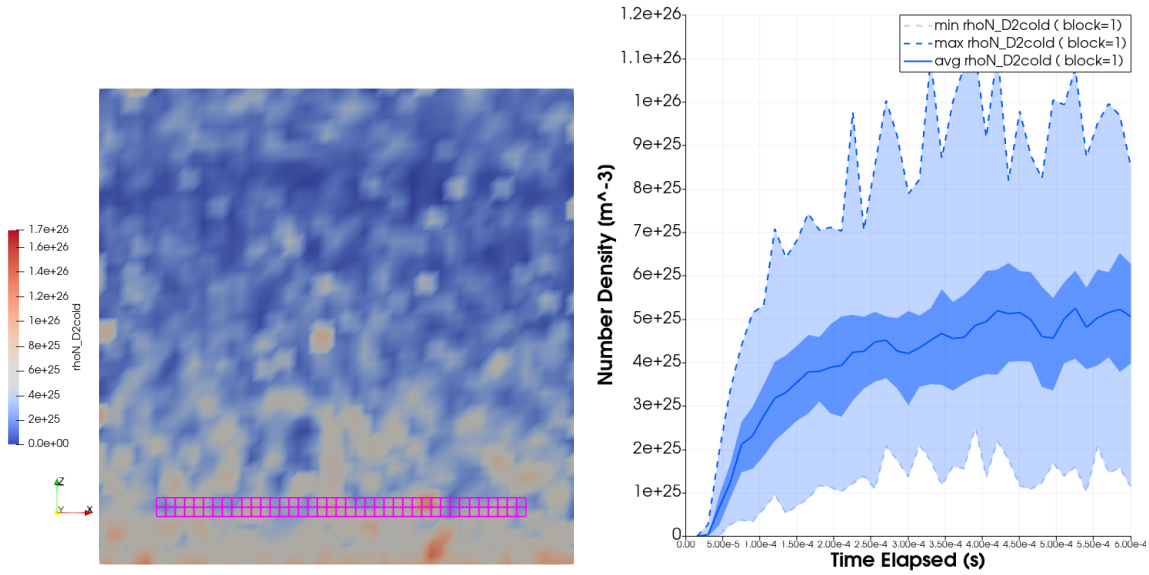
6.4.4 Verification of Simulation Results

Using a very similar argument for the one presented in Section 6.1.4, we can provide another time estimate for the initial density spike in Figure 6.10b. This time, we are setting an upper bound on when we should see the spike in density based on the average inflow velocity c_{avg} that was defined in Chapter 3 in Equation 3.4. This is a good way of setting an upper bound because it is when the first real group of collisionless particles would have gotten there, but with the incorporated velocity bias from the pressure gradient there should be a spike before this upper bound of $50\mu s$.

The value $50\mu s$ comes from the selected cells being centered around $10cm$ from the inlet. This means that with $c_{avg} \approx 1800$, $t = \frac{d}{v} \approx 50\mu s$. In figure 6.10b, the initial density spike is just before $50\mu s$, which verifies that collisionality is impacting the simulation.

6.5 High Pressure Filling in an Approximated PI-3 Geometry

This final simulation is a culmination of the four simulations previously discussed in this section. It is the simulation that this thesis was built around as it provides insight into the dynamics of the neutral gas filling process of General Fusion's PI-3 plasma injector. This case doesn't have a previous



(a) Simulation plot showing cells included in the plot of Figure 6.10b.

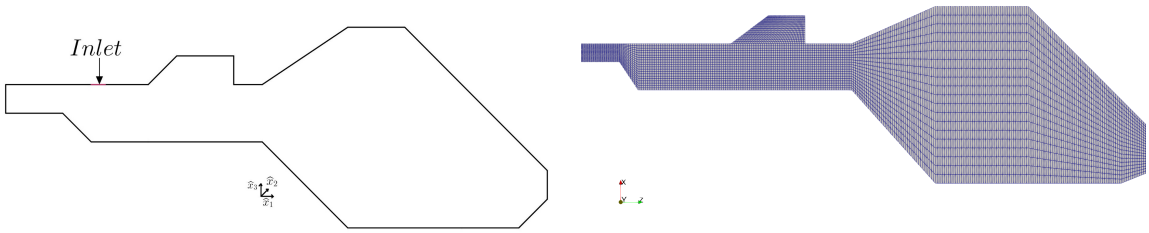
(b) Plot of density over time in the high pressure bottom edge filling simulation, 10cm from the inlet.

Figure 6.10: Plot of density over time, with the cells selected shown on the left in Figure 6.10a and the plot itself shown on the right in Figure 6.10b.

basis to be compared to other than an intuitive understanding of fluid mechanics, verification that the hyStrath software is being utilized correctly (from Sections 6.1, 6.2, 6.3, and 6.4), and the original verification of the hyStrath software [23].

6.5.1 Geometry and Simulation Mesh

The geometry used in this section was based on the same vertices as the geometry in Section 6.5.1, but was made up of slightly different blocks. It also had a more representative inlet, which was approximated to have the same total circular area as the 25 valves on the real PI-3 device.



(a) Sketch of the simulation geometry

(b) Mesh that is generated for the simulation.

Figure 6.11: Diagrams of the simulation geometry used in Section 6.3, where the darker red part of the sketch is the inlet.

6.5.2 DSMC Assumption Verification

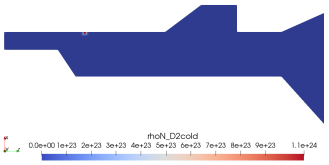
After constructing an appropriate mesh, the assumptions presented in Section 5.4.2 were calculated to see if they held, and the results of those calculations are in Table 6.6.

Table 6.6: DSMC assumptions verification table for low pressure filling in an approximated PI-3 plasma injector.

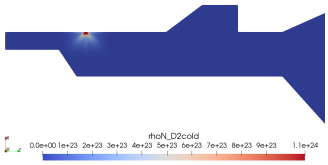
Condition	$\frac{\lambda}{d} \gg 1$	$\frac{n_0}{n_{max}} \gg 1$	$N_{DSMC} > 20$	$\frac{L_{cell}}{d} \gg 1$
Satisfied	Y	Y	Y	Y

6.5.3 Time Evolution of Density in the Entire Domain

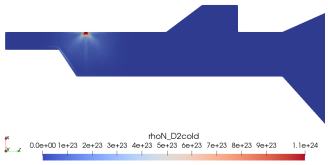
A full animation of this filling process as well as the other filling processes presented in this chapter are available in the GitHub repository discussed in Appendix A. That gives a more full continuous picture of the filling process, but six moments during that simulation are presented in Figure 6.12. This simulation shows that there is higher density accumulation within the Penning trap than the surrounding area which is very positive for plasma formation. However, quite a bit of gas is also lost to the end chamber on the far left of the domain where it is likely electrons will collide with walls before causing significant plasma formation, wasting a portion of the electric potential that forms the plasma.



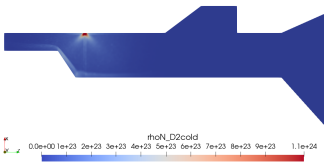
(a) Density distribution at $t = 15\mu s$.



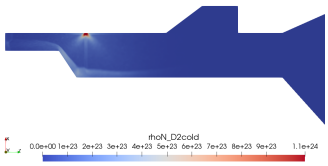
(b) Density distribution at $t = 75\mu s$.



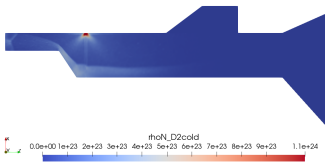
(c) Density distribution at $t = 135\mu s$.



(d) Density distribution at $t = 195\mu s$.



(e) Density distribution at $t = 255\mu s$.

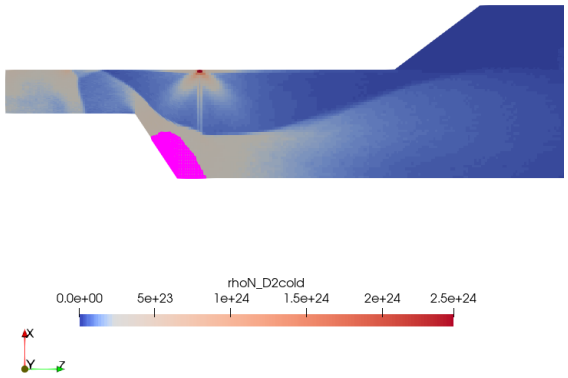


(f) Density distribution at $t = 315\mu s$.

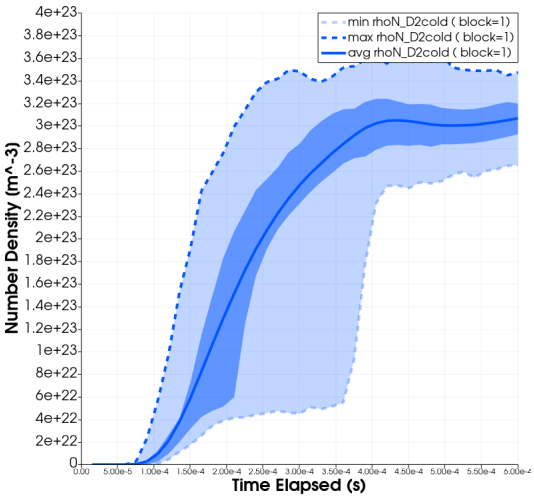
Figure 6.12: Plots of the time evolution of the density distribution for a gas filling process in a cube with an inlet slit. (high resolution animation available on the GitHub from Appendix A)

6.5.4 Time Evolution of Density in the PI-3 Penning Trap

The most interesting point of density evolution in the PI-3 plasma injector is in the Penning trap (see Section 4.1.2 for more information about Penning traps). The Penning trap is where ionization occurs, and that means small density variations in this area can lead to large differences in how much plasma is formed during the PI-3 ionization process.



(a) Simulation plot showing cells included in the plot of Figure 6.13b.



(b) Plot of density over time in the high pressure approximate PI-3 filling simulation, $10cm$ from the inlet.

Figure 6.13: Plot of density over time, with the cells selected shown on the left in Figure 6.13a and the plot itself shown on the right in Figure 6.13b.

The results of this simulation seem to indicate that an optimal time for ionization might be around $300\mu s$, since that is before you get too much particle accumulation just outside the penning trap for effective ionization, but maximum local density within the penning trap itself is almost reached. That is slightly longer than General Fusion is currently running their PI-3 gas inflow at $240\mu s$. I think the continuation of this project into ionization modelling using these simulation results as an input would benefit from trying out various different output times as the base of an ionization simulation to find the most effective gas puff length.

Chapter 7

Conclusion

7.1 Initial Simulations and Research

In a simulation based project, the majority of the work is selecting the correct tools to use. With the correct tool (DSMC in this case) and sufficient computational power (thanks to SciNet access through UTIAS) the problem itself is often reasonably simple. That is why the majority of the body of this thesis is made up of the two literature reviews, and even the simulations of Chapter 6 were primarily test simulations. This background work is the backbone of this thesis, and without it, there is no value in the final simulation.

7.2 Discussion of Final Simulation

The final simulation showed accumulation of gas in the Penning trap of the PI-3 plasma injector, but also showed gas accumulation in other parts of the reactor. It seems like this could be an energy sink stopping more efficient ionization in the Penning Trap, but it is difficult to know without running an ionization model on the results. There was also a steep increase of number density in the penning trap itself until after $300\mu s$, even though General Fusion ionize their gas at $240\mu s$. This should be further explored with experiments at a number of different lengths of filling time or the extension of this project to include an ionization model.

7.3 Future Work

This thesis sets up the continuation of this project in the form of a detailed ionization model. The literature review is already written, so the project should be able to be continued by anyone with some experience or even basic coursework in plasma physics and CFD. At this point the majority of

the work in creating the final plasma model has been done, but a lot of the final benefit and results have not been realized due to the lack of an ionization model to finish off the project.

Bibliography

[1] *Bringing Fusion Energy To Market*. Sept. 2023. URL: https://generalfusion.com/.

[2] Clinton Groth. URL: https://arrow.utias.utoronto.ca/~groth/.

[3] Kelly Epp et al. "Confinement Physics on Plasma Injector 3". In: *APS Division of Plasma Physics Meeting Abstracts*. Vol. 2020. APS Meeting Abstracts. Jan. 2020, ZP07.011, ZP07.011.

[4] John Marshall. "Performance of a Hydromagnetic Plasma Gun". In: *Physics of Fluids* 3.1 (Jan. 1960), pp. 134–135. DOI: 10.1063/1.1705989.

[5] G A Bird. "Monte Carlo simulation of gas flows". In: *Annual Review of Fluid Mechanics* 10.1 (1978), pp. 11–31. DOI: 10.1146/annurev.fl.10.010178.000303.

[6] I. D. Boyd, D. R. Beattie, and M. A. Cappelli. "Numerical and experimental investigations of low-density supersonic jets of hydrogen". In: *Journal of Fluid Mechanics* 280 (1994), pp. 41–67. DOI: 10.1017/s0022112094002843.

[7] June 2021. URL: https://www.iaea.org/bulletin/magnetic-fusion-confinement-with-tokamaks-and-stellarators.

[8] URL: https://www-pub.iaea.org/MTCD/Publications/PDF/te_1466_web.pdf.

[9] URL: https://lasers.llnl.gov/about/how-nif-works.

[10] URL: https://www.britannica.com/science/mean-free-path.

[11] Nishanth Dongari, Ashutosh Sharma, and Franz Durst. "Pressure-driven diffusive gas flows in micro-channels: From the Knudsen to the continuum regimes". In: *Microfluidics and Nanofluidics* 6.5 (2008), pp. 679–692. DOI: 10.1007/s10404-008-0344-y.

[12] I J Hewitt. *Continuum Mechanics*. URL: https://people.maths.ox.ac.uk/hewitt/publications/hewitt_karthaus_continuum_notes.pdf.

[13] Caroline Terquem. *Fluids*. URL: https://www.physics.ox.ac.uk/system/files/file_attachments/notes.pdf.

[14] Israel B. Sebastião, Marat Kulakhmetov, and Alina Alexeenko. "Comparison between phenomenological and ab-initio reaction and relaxation models in DSMC". In: *AIP Conference Proceedings* (2016). DOI: 10.1063/1.4967656.

[15] Iain D. Boyd and Thomas E. Schwartzentruber. "6 - Direct Simulation Monte Carlo". In: *Nonequilibrium Gas Dynamics and Molecular Simulation*. Cambridge University Press, 2017, pp. 183–251.

[16] URL: https://www.openfoam.com/documentation/user-guide/a-reference/a.4-standard-boundary-conditions.

[17] Graeme Austin Bird. *The DSMC method: Version 1.2, 2013*. Createspace, 2013.

[18] Chris Greenshields. *OpenFOAM 1.6 released*. Apr. 2023. URL: https://openfoam.org/release/1-6/.

[19] URL: https://www.openfoam.com/.

[20] hyStrath. *Hystrath*. URL: https://hystrath.github.io/.

[21] Steve Plimpton. *Sparta Direct Simulation Monte Carlo Simulator*. URL: https://sparta.github.io/.

[22] URL: https://www.particleincell.com/starfish/.

[23] C. White et al. "DsmcFoam+: An openfoam based direct simulation Monte Carlo Solver". In: *Computer Physics Communications* 224 (2018), pp. 22–43. DOI: 10.1016/j.cpc.2017.09.030.

[24] URL: https://titan.triumf.ca/equipment/penning_trap/.

[25] Helmut Föll. URL: https://www.tf.uni-kiel.de/matwis/amat/elmat_en/kap_2/backbone/r2_4_2.html.

[26] D. Lumma, J. L. Terry, and B. Lipschultz. "Radiative and three-body recombination in the alcator C-mod divertor". In: *Physics of Plasmas* 4.7 (July 1997), pp. 2555–2566. DOI: 10.1063/1.872234.

[27] David Tskhakaya. "The Particle-in-Cell Method". In: *Computational Many-Particle Physics*. Ed. by H. Fehske, R. Schneider, and A. Weiße. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 161–189. ISBN: 978-3-540-74686-7. DOI: 10.1007/978-3-540-74686-7_6. URL: https://doi.org/10.1007/978-3-540-74686-7_6.

[28] F. Collino, T. Fouquet, and P. Joly. "Conservative space-time mesh refinement methods for the FDTD solution of Maxwell's equations". In: *Journal of Computational Physics* 211.1 (Jan. 2006), pp. 9–35. DOI: 10.1016/j.jcp.2005.03.035.

[29] URL: https://smileipic.github.io/Smilei/index.html.

[30] TFDzarm. *TFDzarm/Picfoam*. URL: https://github.com/TFDzarm/picFoam.

[31] Christoph Kühn and Rodion Groll. "Picfoam: An openfoam based electrostatic particle-in-cell solver". In: *Computer Physics Communications* 262 (May 2021), p. 107853. DOI: 10.1016/j.cpc.2021.107853.

[32] URL: https://osiris-code.github.io/.

[33] URL: https://cccbdb.nist.gov/exp2x.asp?casno=7782390.

[34] URL: https://ifd.ethz.ch/research/group-jenny/projects-gas-kinetics/rarefied-gas-simulations.html.

Appendix A

GitHub Repository

Link to GitHub Repository: *here*

Certain resources associated with this document are more effectively delivered in formats other than a traditional written text thesis. Examples of this are animations of density variations over time in the entire simulation domain, and configuration files that could be used to reproduce results. These resources have been kept in their original form and are available in the GitHub repository linked at the beginning of Appendix A.

This means that the appendices for this document are very short, as any additional plots and other information are available on the GitHub.

Appendix B

Tutorial and Software Download Links

Here are links that contain all the different software packages tested or used in this thesis.

B.1 DSMC Software

Downloading Bird's Software: software here

Downloading OpenFOAM: tutorial here

Downloading hyStrath: tutorial here

Downloading SPARTA: software here

Downloading Starfish: software here

B.2 PIC Software

Downloading hyStrath: tutorial here

Downloading Smilei: software here

Downloading Starfish: software here

Downloading picFOAM: software here

Downloading OSIRIS: software here

B.3 Visualization Software

Downloading ParaView: software here