Modeling the Fusion Reaction in an Inertial Electrostatic Confinement Reactor with the Particle-in-Cell Method

by
Jacob van de Lindt

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Nuclear Engineering and Physics
(Honors Scholar)

Presented May 29, 2020
Commencement June 2021

AN ABSTRACT OF THE THESIS OF

Jacob van de Lindt for the degree of Honors Baccalaureate of Science in Nuclear Engineering and Physics presented on May 29, 2020. Title: Modeling the Fusion Reaction in an Inertial Electrostatic Confinement Reactor with the Particle-in-Cell Method.

Abstract approved: Brian Woods

Nuclear fusion has the potential to meet the energy needs of our civilization for centuries to come and does not produce harmful greenhouse gasses or produce long-lasting radioactive waste. A fusion reaction can be readily demonstrated in a confinement scheme known as inertial electrostatic confinement (IEC) fusion. In this work, an IEC device is proposed that can perform a study of how the fusion rate in an IEC depends on cathode voltage. A 2D simulation code is written in python to model the device and predict the fusion response to grid voltage at low and high voltages. The simulation is based on the particle-in-cell (PIC) method for modeling plasmas. The fusion reaction was modeled in the simulation by applying the 5-parameter-fitting method for nuclear fusion cross sections. A formula for the probability of a fusion reaction occurring per simulated particle path length is proposed based on physical and numerical parameters. The simulation demonstrates that as the cathode grid voltage is increased, the fusion rate occurring in the device increases in the same manner as key IEC experiments. The seldom-studied start-up behavior of the IEC was resolved in the simulation implemented here and presented for different cathode voltages.

Key Words: Fusion, particle-in-cell, inertial electrostatic confinement

Corresponding e-mail address: vandelij@oregonstate.edu

Modeling the Fusion Reaction in an Inertial Electrostatic Confinement Reactor with the Particle-in-Cell Method

by
Jacob van de Lindt

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Nuclear Engineering and Physics
(Honors Scholar)

Presented May 29, 2020
Commencement June 2021

Honors Baccalaureate of Science in Nuclear Engineering and Physics project of Jacob van de Lindt presented on May 29, 2020.

APPROVED:

_____

Brian Woods, Mentor, representing the Department of Nuclear Engineering

_____

Janet Tate, Committee Member, representing the Department of Physics

_____

Ethan Minot, Committee Member, representing the Department of Physics

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Jacob van de Lindt, Author

Table of Contents

Table of Figures

Chapter 1 – Introduction

1.1 Motivation and Objective

Climate change and humanity's ever-increasing demand for energy sets up nuclear fusion as an enticing source of clean energy. If it were to work at levels anticipated, it would be a more reliable source of energy than wind and solar, and a cleaner energy source than fossil fuels and nuclear fission. The leading method to attempt to harness a fusion reaction for energy is a tokamak: a large donut-shaped machine that utilizes magnetic fields to confine a hot plasma long enough for it to produce energy. The tokamak quickly becomes large, expensive, and complex. An alternative to the tokamak which can readily produce nuclear fusion is the inertial electrostatic confinement (IEC) device [1]. At present, IECs are not contenders for a fusion power device, but due to their comparative ease at accomplishing fusion they do allow for the study of fusion reactions. It is the purpose of this research to propose a design for an IEC fusion device to study the fusion reaction's response to the IEC input voltage, and model the processes occurring within the device by utilizing computational methods. By developing a computational method that can model the small IEC device created here, the computations can be extended to simulate bigger IEC devices with variable geometries and power levels.

1.2 Nuclear Fusion

Nuclear fusion is the process by which small elements and isotopes can combine into heavier isotopes. The fusion reaction products will have a net mass that is lower than the reactants that entered the fusion reaction, called the mass defect. This mass defect Δm is exactly the amount of kinetic energy gained by the products of the fusion reaction. This kinetic energy gain can be solved for using Einstein's famous equation

$$Q = \Delta m c^2 , \tag{1.1}$$

where Q is the kinetic energy gained by the products and c is the speed of light in vacuum. Fusion reactions occur between two positively charged nuclei only when the two reactants come close enough for the strong force to fuse the two reactants. The two reactant nuclei must overcome the powerful Coulomb repulsion between the two positive nuclei for the fusion reaction to occur. The energy to overcome this Coulomb barrier can be supplied by the temperature of the products. If the thermal motion of the reactants

is high enough, then the reactants will have enough kinetic energy to overcome the Coulomb barrier and initiate the fusion reaction. It is the high temperature requirements of fusion reactants that necessitate the merging of fusion science and plasma physics: most common fusion reactants, deuterium and tritium, exist in a plasma state at the temperatures required for fusion. The main challenge of fusion energy science is electromagnetically confining this hot plasma long enough for fusion to occur while producing electrical energy greater than the energy needed to confine the plasma. Such a confinement configuration still eludes humanity to this day.

The fusion reaction occurring in the device proposed here is D-D (deuterium-deuterium) fusion. An extremely small amount of D-T (deuterium-tritium) fusion could occur as tritium is a product of the D-D reaction. However, D-T fusion is likely not occurring at appreciable rates due to the low amount of D-D fusion occurring in the device and the short operating times. The branching ratio, defined as the probability of a certain set of products emerging from the same fusion reaction, is 50% for D-D products [2]. The two D-D reactions and the D-T reaction are

$$ {}_1^2D + {}_1^2D \rightarrow {}_1^3T + {}_1^1H, \qquad Q = 4.03 \quad \text{MeV} \tag{1.2} $$

$$ {}_1^2D + {}_1^2D \rightarrow {}_2^3He + n, \qquad Q = 3.27 \quad \text{MeV} \tag{1.3} $$

$$ {}_1^3T + {}_1^2D \rightarrow {}_2^4He + n, \qquad Q = 17.58 \ \text{MeV} \tag{1.4} $$

where *n* is a neutron. The kinetic energy gain Q is shared between the products of the fusion reaction, with the lighter product particle receiving a larger proportion of the energy. The neutron produced by the D-D reaction has an energy of 2.45 MeV, and the proton produced by the D-D reaction has an energy of 3.02 MeV [2]. These energy values are important for particle detection, which is discussed in Section 2.4. The neutron produced by the D-T reaction has an energy of 14.1 MeV. It is the 2.45 MeV neutron that can be measured to validate that fusion is occurring in the proposed device. The number of 2.45 MeV neutrons detected per second can be used to predict the total fusion rate occurring in the device, and can be used to validate the output of the computational model that was developed.

1.2.1 Fusion Cross Sections

Deuterium ions can interact with one another via scattering or by nuclear fusion. A scattering event is much more likely to occur than a fusion interaction. Figure 1.1 compares the fusion cross sections for various species of reactants.

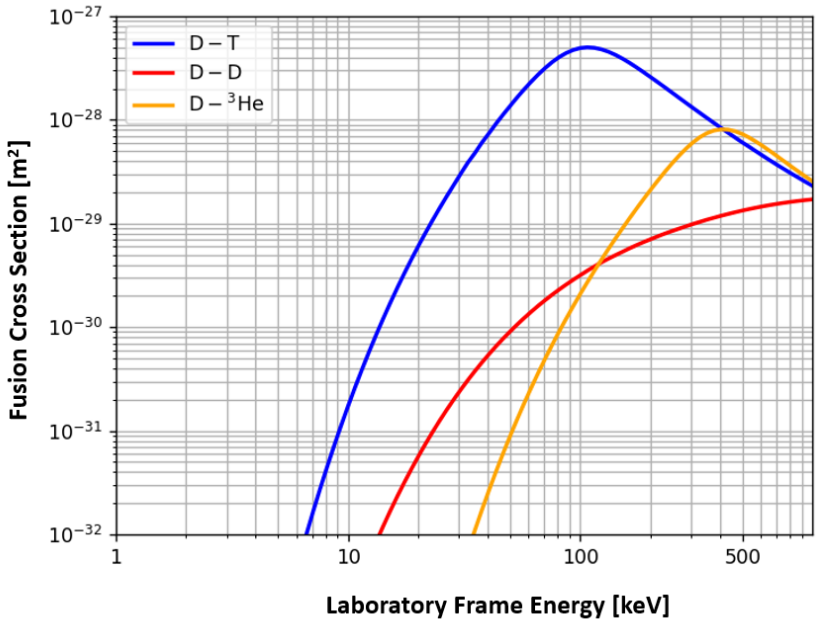


Figure 1.1 Cross sections for fusion reactions versus laboratory frame energy. In this experiment, D-D reactions are the primary fusion event occurring in the device. The cross section for D-D is shown as the red curve. The laboratory energy can roughly be thought of as the required grid voltage to achieve a certain fusion cross section.

Microscopic cross section σ has units of area and can be thought of classically as the planform area of a target particle. Thus, the larger this area, the higher the probability a projectile particle will collide with the target. This classical thinking breaks down in that the cross section is a quantum phenomenon. The cross section of a particle depends on what type of interaction is occurring, the energy of the interacting particles, and is purely statistical. This energy dependence of microscopic cross section is shown above in Figure 1.1 for fusion reactions of various reacting species. A more useful metric is the macroscopic cross section $\sum$, which is defined below in Equation 1.5 for the fusion reaction (denoted by the subscript f).

$$\sum_f = \sigma_f n_{ion} \tag{1.5}$$

The macroscopic cross section has dimensions of per distance and can be thought of as the probability of an interaction occurring when multiplied by a particle path length. The microscopic fusion cross section for the D-D interaction, as shown by the red curve in Figure 1.1, is energy dependent. The macroscopic cross section also depends on the ion density n_{ion}, which in general is a function of position.

1.2.2 Inertial Electrostatic Confinement Fusion

Inertial electrostatic confinement fusion does not rely on any magnetic field, and instead uses an electric field to perform the plasma confinement. The device is formed by arranging two spherical metal meshes concentrically and putting an extremely large voltage across them, typically 10 to 100 kV in vacuum (Figure 1.2). The meshes are supported by a high-voltage stalk (Figure 1.3). Fusion fuel, in this case deuterium, is then released into the chamber where the large voltage ionizes a portion of the gas. These positively charged ions are accelerated by the electric field toward the center of the device, where most will pass straight through the cathode at the center and out the other side due to the large gaps in the metal mesh. Once on the other side, the electric field slows the ions and eventually turns them back towards the center, setting up a recirculation. The ion density in the center increases, and eventually fusion occurs. This does not require nearly as large a machine as the tokamak: typical IEC devices comfortably sit on tabletops [1]. In this work, a hypothetical IEC device is proposed that could demonstrate fusion occurring, and a working computational model of the device's performance is created.

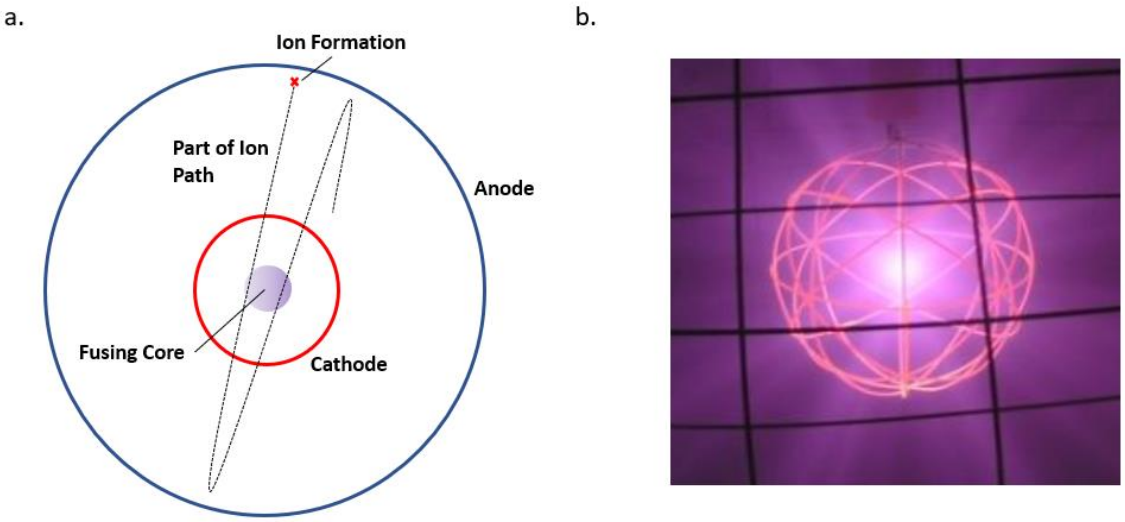


Figure 1.2 a) A general schematic of an idealized IEC fusion device. The red spherical cathode has a large negative voltage, while the larger blue sphere is a grounded anode. Deuterium is ionized near the anode, and these positive ions are accelerated by the radial electric field toward the center. The ions pass through the center and out the other side, where the electric field slows and turns them back. A high-density core sets up due to the presence of recirculating ions, and fusion occurs. b) A photo of an IEC device in operation at the University of Wisconsin-Madison [3]. The cathode grid glows orange from heat, while the anode (foreground) does not. Reproduced with permission.

1.3 Current and Future Application of Inertial Electrostatic Confinement

Although IEC fusion is not yet seen as a viable source for fusion electric power reactors, they have been deemed useful in several other applications.

Neutrons are used in neutron activation analysis, where materials are characterized to determine what elements make up the material sample [1]. This is accomplished by bombarding a material sample with neutrons and observing the type of radiation that is emitted from the material by isotopes formed in the bombardment. This emitted radiation will have characteristics unique to certain isotopes. Thus, the material can be characterized. A problem arises, however, in that most neutron sources are themselves simply decaying radioactive isotopes, which pose both health and proliferation risks because they cannot be shut down. An IEC device can produce a large amount of neutron radiation on demand and can be shut off simply by cutting power to the device when neutron radiation is no longer needed, making the IEC an attractive neutron source for neutron activation analysis.

IEC devices have also been proposed as a space propulsion concept [1], where argon fuel is injected into an IEC device and guided out the back of the device to form plasma thrust. This device could have a higher specific impulse than other well-established plasma thrusters such as the Hall thruster [1].

1.4 Plasma Physics

A plasma is a collection of particles in which some or all the particles carry a net charge. These charged particles, or ions, are accompanied by a sea of free electrons. The net electron charge typically equals the net ion charge at large length scales. This charge equality is termed quasineutrality. Plasma physics is the study of ionized gases which exhibit quasineutrality at length scales larger than a length known as the Debye length,

$$\lambda_D = \sqrt{\frac{\varepsilon_o k_B T_e}{n_o q_e^2}}\,. \tag{1.6}$$

Here ε_o is the permittivity of free space, k_B is Boltzmann's constant, T_e is the electron temperature, q_e is the charge of an electron, and n_o is the charge density (number of charges per unit volume) of either electrons or ions. Quasineutrality means that the ionized plasma will appear neutral at length scales larger than the Debye length. The Debye length can be thought of as the distance needed to shield an external

charge from the rest of the plasma. The electron's parameters appear in Equation 1.6 due to the electron's higher mobility within a plasma than the ions. This mobility can be readily explained by the fundamental Lorenz force law, which governs plasma behavior. The Lorenz force law states that

$$m\frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \tag{1.7}$$

Here m is the mass of the particle, **v** is the particle velocity, **E** is the electric field, q is the particle charge, and **B** is the magnetic field. The ion has a mass several orders of magnitude larger than an electron, so in an identical external **E** and **B** field, a particle will experience an acceleration proportional to its charge to mass ratio $\frac{q}{m}$, which for an ion will be many orders of magnitude smaller than an electron. Therefore, when an external charge is introduced to a plasma, it is the electrons which accelerate to shield the external charge from the rest of the plasma. Hence, the Debye length is a function of electron parameters only.

Plasmas share many of the same attributes as gasses, although what sets them apart is that longer range interparticle Coulomb forces cannot be ignored. Plasmas can exhibit collective behavior due to the Coulomb interactions between plasma particles, and between the plasma and external electromagnetic fields applied to the plasma. This behavior enables plasmas to be engineered and controlled by an applied electromagnetic field.

1.4.1 Plasma Modeling

Plasma modeling shares many similarities to fluid modeling. One of the main similarities is that both can utilize a numerical mesh grid to discretize the problem space. The advantage of this method to solving problems is that a complex system of differential equations can be broken down into a large system of linear difference equations that can be readily solved by a computer.

The density of a plasma plays a crucial role in its behavior and how it can be analyzed. A low-density plasma can in some instances be modeled as a set of charged particles that do not interact with each other at all (collisionless, noninteracting), and instead only move due to an external electromagnetic field. As the plasma density increases, the electromagnetic field generated by the plasma particles themselves can no longer be ignored, and the complexity of the plasma behavior increases drastically because each particle is influenced by both the external electromagnetic field and the electromagnetic fields of all the

other plasma particles. When the plasma density is increased further, the behavior begins to have fluid characteristics due to particle collisions, and principles from statistical mechanics can be applied to the plasma. By taking moments of the Boltzmann Equation, both a two-fluid treatment and a single fluid treatment (magnetohydrodynamics) can be derived [4].

In this study, the plasma has too low a density for the Boltzmann Equation to apply and too high a density to ignore interparticle Coulomb interactions. A middle-of-the-road treatment was taken up, where a particle approach was utilized to analyze the plasma's response to an external electric field with interparticle Coulomb interactions included. Although particle collisions were ignored in the kinetic behavior of the plasma, collisions cannot be ignored entirely in this study because they are required for nuclear fusion to occur in the plasma.

A computational method which can model a self-interacting low-density plasma is the particle-in-cell (PIC) technique, explained in detail in Section 2.1.3. The PIC method does not, however, include particle collisions necessary to study fusion. To handle collisions, the PIC method was modified here by utilizing fusion cross sections (cross section implementation discussed in section 2.1.4).

1.5 Construction Considerations

There are several important design principles that go into creating an IEC fusion device. In an ideal IEC, the anode would be a completely transparent sphere with a potential of 0 V, and the cathode would be a smaller completely transparent sphere with a potential << -100 kV. This would set up a perfectly radial electric field pointing to the center of the device. These two spheres would be fixed in space in a large vacuum. The ions would circulate until they fuse, and the ions would ideally not be lost from the system by any other mechanism.

This idealization is first degraded by the presence of physical metal grids (Figure 1.2 b). These grids are not completely transparent, so ions collide with them while the ions undergo their recirculation. The ions are lost to the plasma when such collisions with the grid occur. The grids are also not perfectly spherical shells, but are instead discrete wires of potential with large gaps of vacuum between wires. Thus, in general the goal of anode and cathode wire grid design is to maximize the uniformity of the potential distribution of the grid sphere, while at the same time maximizing the transparency of the grid. These two constraints are competing, and in general transparency takes the higher priority: large gaps in the

spherical surface are formed between a relatively small amount of wire to prevent ion collisions with the wires.

The wire grids cannot be levitated without a support structure, and the grid potential cannot be set without a connecting wire. These two issues are remedied by a high voltage stalk, which is a cylindrical tube with wire running up the center, shown in Figure 1.3. The stalk is typically made from ceramic to withstand the high temperature of the plasma and the ion bombardment the stalk will endure. The presence of this high voltage stalk adds an asymmetry to the electric field due to both the high voltage wire running through it, and the loss channel the ceramic tube represents to the recirculating ions. When designing a high voltage stalk, it is desired to minimize the disturbance the stalk will cause to the electric field, and to reduce the diameter of the stalk to avoid disturbing ion recirculation paths.

The IEC vacuum chamber is of finite size and thus will interact with ions and form another loss channel to the system. The vacuum chamber proposed for this study is made from aluminum and thus will interact with the electric field created by the grids. The vacuum chamber is grounded with the same potential as the anode wire grid. In an ideal situation, the vacuum chamber would be spherical as well, and concentric with the grids to minimize its influence on the electric field. Spherical vacuum chambers are expensive, however, so a cylindrical vacuum chamber was found to be a good compromise in this study.

Lastly, the power supply to the cathode would be ideally able to supply an extremely large negative potential. This is because the ability of an IEC to perform D-D fusion is directly related to its maximum potential drop from the anode to the cathode. Figure 1.1 shows that as the ion energy in the laboratory frame increases, so does the fusion cross section. This energy in an IEC is supplied by the grid voltage; a higher grid voltage is required to achieve a higher ion kinetic energy. Thus, increasing the grid voltage increases the fusion cross section and results in a higher D-D fusion reaction yield. A typical IEC that can be physically constructed is shown below in Figure 1.3.

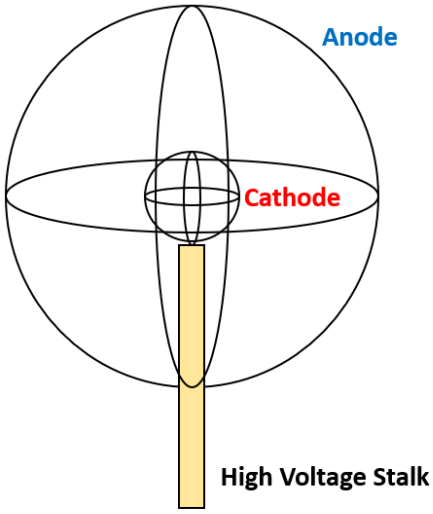


Figure 1.3 Realistic schematic of IEC device. The high voltage stalk supports the anode and cathode. The anode and cathode are far from being an ideal spherical surface, and are instead constructed from several wires. The stalk and wires are loss channels for ions recirculating within the device and inhibit ideal IEC operation.

Chapter 2 – Methods

The study of nuclear fusion in IEC devices involves a variety of disciplines ranging from plasma physics, kinetics, statistical mechanics, electromagnetism, and nuclear physics. In this chapter, the methods used to combine these diverse disciplines will be outlined. The computational domain and its simplifications will be introduced first, followed by an explanation of the particle-in-cell (PIC) method. Parameters pertinent to fusion reactions in the model will be introduced. This will be followed by an overview of the proposed physical system and will conclude with a discussion of neutron detection in the context of this experiment.

Neutron count versus applied grid voltage is the main deliverable produced in this study. It can be used to demonstrate that fusion is being achieved within the proposed device. Neutron count versus applied grid voltage is additionally used as the initial validation for the computational methods being developed. The PIC model determines the ion density and fusion rate occurring within the device, and thus does not directly predict the neutron count that would be seen by the neutron sensor discussed in Section 2.4. Instead, the fusion rate found by the PIC model could be treated as an isotropic neutron source. This isotropic neutron source and the distance to the neutron detector can be used to back out a computationally determined neutron count rate given the inputted grid voltage.

2.1 Computational Domain

The computational model of the IEC device replaces the physical wire grids with two circular equipotentials of the same radii. The potential of these circles is equal to the grid voltage. The computational domain is two-dimensional and extends past the outer grid to the radius of the vacuum chamber, shown in Figure 2.1. A major source of error with this method is assuming the anode and cathodes are closed circles, when in reality they are physical spheres made of discrete wires with the majority of the surface area being empty space (Figure 1.3). This discrepancy between model and reality is remedied by assigning a geometric transparency to the anode and cathode circle models. The idea of geometric transparency is discussed further in section 2.1.3.5 and is effectively the probability that an ion will collide with the anode or cathode. Geometric transparency will fail to capture interesting grid effects such as the formation of ion beams and various well-known modes of IEC operation such as star mode and jet mode [1].

2.1.1 Defining the Geometric Mesh

The computational domain was meshed using a regular Euclidian mesh with equal cell dimensions in *x* and *y*, despite the spherical geometry of the device. The decision to use a Euclidian mesh as opposed to a spherical mesh is due to the vacuum chamber not respecting spherical symmetry. The domain's outer-most boundary is the cylindrical vacuum chamber. The stalk, discussed in Section 1.5, was ignored in this study, which likely adds a considerable source of error in the simulation due to the stalk playing a major role as an ion loss channel. The computational domain used in this model is visualized in Figure 2.1.

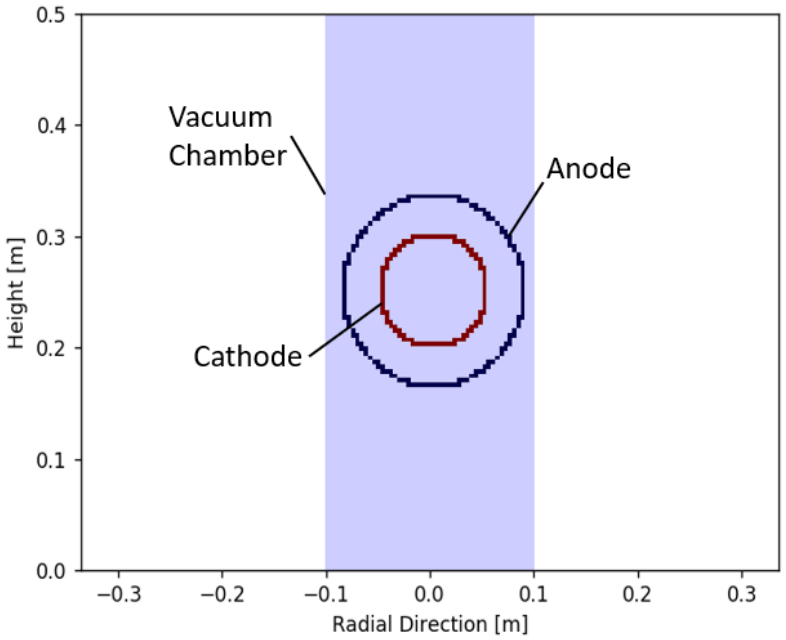


Figure 2.1 Shown here is the physical domain's representation as a computational grid. The light blue cells represent cells that are within the vacuum chamber, the red cells represent the cathode and the dark blue cells represent the anode.

The regular Euclidian mesh making up the computational domain was extended over a region with N_x cells in *x* direction, and N_y cells in *y* direction. This domain can be translated into a set of N_xN_y equations, one for each cell corner by using the discretized Poisson's Equation:

$$\nabla^2\phi = -\frac{\rho}{\varepsilon_o}\,. \tag{2.1}$$

Here ρ is the net charge density, given by $q(n_{ion} - n_e)$, the number density of ions minus the number density of electrons times q, the charge of a proton. ϕ is the electric potential, and ε_o is the permittivity of free space. This equation can be discretized in a finite difference form in three dimensions:

$$\frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{(\Delta x)^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{(\Delta y)^2} + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{(\Delta z)^2} = q\frac{n_e - n_i}{\varepsilon_o}. \quad (2.2)$$

Equation 2.2 in two dimensions is the basis for the computational simulation carried out in this work.

2.1.2 Initial Conditions and Boundary Conditions

At the radius of the cathode (red cells in Figure 2.1), an equipotential circle is defined with a Dirichlet boundary condition equal to the grid voltage imparted to the device. A similar condition is applied to the anode (dark blue cells), which is set to 0 V. These circles are given a geometric transparency discussed in section 2.1.3.5. The vacuum chamber walls are simulated as an open boundary: any particle that collides with this surface passes through and is lost from the computational domain. The domain is initiated with zero ions present, with a particle source near the anode that is discussed in further detail in Section 2.1.3.5. Particles are assumed to have no net initial velocity beyond thermal motion sampled from a Maxwellian distribution of initial ion velocities.

2.1.3 Particle-in-cell Technique

At the low densities occurring in the IEC device studied here, the plasma behaves more like a collection of individual particles than a fluid. However, in addition to interacting with the external electric field supplied by the wire grids, the particles are also influenced by the electric field from the other particles in the system. The electric field from particle *i* on particle *j* for a point charge (ion or electron) is given as

$$\mathbf{E_{ij}} = \frac{1}{4\pi\varepsilon_o}\frac{q_i}{r^2}\mathbf{r_{ij}}, \quad (2.3)$$

where $\mathbf{r_{ij}}$ is a unit vector pointing from particle *i* to particle *j*, *r* is the distance between particle *j* and particle *i*, and q_i is the charge of particle *i*. In theory, determining the net electric field on particle *j* would involve summing the vector contributions over all *i*. This leads to an n^2 problem: the electric field felt by one particle is determined by knowing the electric field contribution from every other particle in the system. This is computationally taxing for even a low number of particles. The concept of a Debye length,

defined in Equation 1.6, can be used to reduce this complexity by realizing that particles more than one Debye length from the particle of interest will not contribute to the electric field felt by the particle of interest due to Debye shielding.

The PIC method starts by discretizing the geometry as mentioned in Section 2.1.1 into cells, which are surrounded by four nodes (corners of cells). Properties such as charge density, electric potential, and electric field exist only at the cell nodes. Individual particles during each iteration populate the cell, but their charge is averaged to the cell nodes during each iteration to get field properties such as potential and density.

The side length of the cells are Δx and Δy. By setting both Δx and Δy to at minimum the Debye length λ_D, the electric field felt by particles within a cell of interest is guaranteed to come from the electric field at the nodes immediately surrounding that cell only. This is because all other nodes are further than one Debye length away as shown in Figure 2.2, and thus these nodes are shielded from the perspective of particles within the cell of interest.

Even when reducing the number of particles to only those within one Debye length cell, there are still too many individual ions to model with reasonable speed. To avoid simulating a vast number of individual ions, the PIC method utilizes many *macroparticles* [5]. A macroparticle simply represents the mass and charge of many thousands or millions of individual ions combined into one point. The ratio of physical ions per macroparticle is termed the specific weight, w_s.

The PIC algorithm in this study utilizes the following steps in the main loop [5]:

1. Determine the Charge Density
2. Determine the Electric Potential
3. Determine the Electric Field
4. Move the Macroparticles and Check for Fusion
5. Sample Sources and Determine Losses
6. Repeat until Maximum Number of Time Steps is Reached

Each of the above steps will be explained in detail in the following subsections.

2.1.3.1 Determine the Charge Density

The charge density is determined only at the node points in the discretized problem domain (corners of cells). A macroparticle's position is recorded and its charge is distributed to the corners of the cell it is in. The largest portion of its charge is given to the nearest node, as shown below in Figure 2.2. The proportion received by each node is determined by the area ratios shown in Figure 2.2. For example, the grey particle is closest to the green node, so this node will receive the largest proportion of the grey particle's charge. If the grey particle has a charge q, then the green node will receive a charge equal to the green area's portion of the entire cell area times q, given by

$$q_G = q\,\frac{(h_y)(h_x)}{\Delta x \Delta y}\,. \tag{2.4}$$

Here q_G is the charge contribution the green node receives from the grey macroparticle, h_x is the grey macroparticle position from the left of the cell, and h_y is the grey macroparticle position from the bottom of the cell.

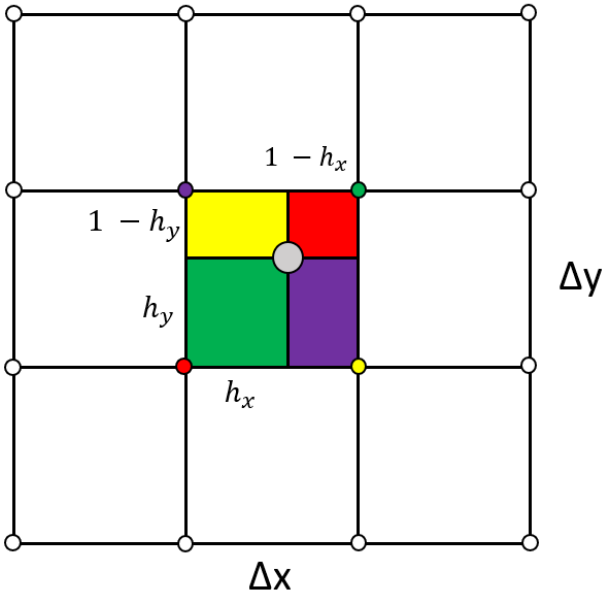


Figure 2.2 Visualizing the charge weighting to nearby numerical grid intersections. The grey macroparticle's charge is distributed to the nodes of the cell it is in. The proportion of charge that is received by the nodes is found by the area ratio using the particle's position in the cell as shown by Equation 2.4. This is visualized by the colored rectangles within the cell. The green node, for example, receives the green rectangle's proportion of the grey macroparticle's charge. Figure based on a discussion in [5].

Each cell in the problem space is looped over and the charges of all the macroparticles within the cell are averaged to the cell nodes, determining the ion charge density n_{ion} at the nodes. The macroparticle positions and velocities are stored.

2.1.3.2 Determine the Electric Potential

An important assumption that will be used in this study is that the electrons in the IEC plasma move instantly relative to the ions. The justification for this assumption was given in Section 1.4, which demonstrated that in an electromagnetic field a particle's acceleration is proportional to the charge to mass ratio $\frac{q}{m}$. The deuterium mass is nearly 4,000 times larger than the electron mass, and hence to the ion the electron moves nearly instantly. The electrons are assumed to be a fluid with a temperature T_e. The electron density is assumed to maintain a Boltzmann distribution given by

$$n_e = n_o e^{\frac{q(\phi - \phi_o)}{k_B T_e}}, \tag{2.5}$$

where ϕ_o is the reference ground potential, 0 V. Equation 2.5 can be substituted into Equation 2.2 for the electron number density in two dimensions with the cell length in the x and y direction equaling the Debye length, yielding Equation 2.6:

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{(\lambda_D)^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{(\lambda_D)^2} = \frac{q}{\varepsilon_o}\left[n_o e^{\frac{q(\phi_{i,j} - \phi_o)}{k_B T_e}} - n_{ion}\right]. \tag{2.6}$$

Equation 2.6 is no longer linear due to the exponential on the RHS containing the potential $\phi_{i,j}$, so a simple matrix inversion by gaussian elimination cannot solve this system of equations. The potential $\phi_{i,j}$ is solved for by setting the domain boundary conditions: the vacuum chamber walls and the anode are set to a potential of 0 V. The cathode is set to the cathode voltage (between -10 and -200 kV).

Next, the interior cells of the domain (excluding the vacuum chamber walls) are looped over and the potential at each cell is updated. The update formula is found by rearranging Equation 2.6:

$$\phi_{i,j}^{k} = \frac{\phi_{i-1,j}^{k} + \phi_{i+1,j}^{k-1} + \phi_{i,j-1}^{k-1} + \phi_{i,j+1}^{k}}{4} - \frac{{\lambda_D}^2 q}{4\varepsilon_o}\left[n_o e^{\frac{q(\phi_{i,j}^{k-1} - \phi_o)}{k_B T_e}} - n_{ion}\right]. \tag{2.7}$$

Here the k^{th} iteration appears as a superscript in the above terms. The domain is looped through row by row as opposed to column by column, explaining why some of the terms in Equation 2.6 have k-1 superscripts while others have k superscripts. Note Equation 2.7 shows that $\phi_{i,j}^{k}$ depends on the cells around it at the current and previous iterations, as well as $\phi_{i,j}^{k-1}$: the potential at the cell whose potential is being solved for. This cell self-dependence comes from the non-linear term introduced by the electron density (Equation 2.5).

To implement the interior anode and cathode potential boundary conditions, the potentials are re-set to their known values between every k-1 and the k^{th} iteration.

After M iterations, the iteration in k is stopped and the final domain potential $\phi_{i,j}^{M}$ is used as the system potential at that specific timestep, timestep s. Determining a good M to use can be done by stopping the iteration in k when the maximum error between $\phi_{i,j}^{k}$ and $\phi_{i,j}^{k-1}$ is below a certain error tolerance ξ :

$$\xi < \max\left|\phi_{i,j}^{k} - \phi_{i,j}^{k-1}\right|. \tag{2.8}$$

In practice, M = 600 was found to guarantee good potential convergence without unreasonably long computational wait times.

2.1.3.3 Determine the Electric Field

Once the electric potential at the nodes is determined using the method explained in Section 2.1.3.2, the electric field at the nodes can be found by finite differencing the electric potential. Along the boundaries, this takes the form of

$$E_{x,ij} = -\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \tag{2.9}$$

$$E_{y,ij} = -\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}, \tag{2.10}$$

and in the interior as

$$E_{x,ij} = -\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, \tag{2.11}$$

$$E_{y,ij} = -\frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}. \quad (2.12)$$

2.1.3.4 Move the Macroparticles and Check for Fusion

The macroparticles are moved using Newton's Second Law according to Equation 2.13:

$$\mathbf{v}^{s+0.5} = \mathbf{v}^{s-0.5} + \frac{q}{m_m}\mathbf{E}\Delta t. \quad (2.13)$$

Here the superscript s refers to the s^{th} timestep iteration. The particle position $\mathbf{r}$ at the next timestep is determined from the s^{th} timestep iteration as

$$\mathbf{r}^{s+1} = \mathbf{r}^{s} + \mathbf{v}^{s+0.5}\Delta t. \quad (2.14)$$

During each ion's motion, the ion has a probability to undergo a fusion reaction. This probability can be expressed using the concept of nuclear fusion cross sections, which are discussed in more detail in Section 2.1.4. The probability P that a macroparticle moving from $\mathbf{r}^s$ to $\mathbf{r}^{s+1}$ will undergo a fusion reaction is given by

$$P = w_s|\mathbf{v}^{s+0.5}\Delta t|\sigma_f(\mathbf{v})n_{ion}. \quad (2.15)$$

Here $\sigma_f(\mathbf{v})$ is the D-D fusion microscopic cross section, which is a function of the particle velocity $\mathbf{v}$. The specific weight w_s is used here because the cross section is for a single ion, and there are w_s ions per macroparticle, all of which could undergo fusion. Cross sections were introduced in Section 1.2.1 and their implementation is discussed in Section 2.1.4. To simulate P, a random number R between 0 and 1 is generated during every particle position update, and if $R < P$, the particle is deleted from the problem domain and the time and location of the fusion reaction is stored. If $R > P$, the particle simply remains at its position.

2.1.3.5 Sample Sources and Determine Losses

After the macroparticle positions are updated, the particle boundary conditions of the problem are applied to see if the macroparticles have left the problem space and need to be deleted from the domain. If the

particle position is outside of the vacuum chamber absorbing wall boundary, then the macroparticle is deleted.

In the IEC, ions can also be lost by colliding with the metal wires that form the wire grid. In the computational model, this is simulated by applying a geometric transparency to the anode, $G_{t,a}$, and to the cathode, $G_{t,c}$ which are defined as

$$G_{t,a} = \frac{\text{surface area of metal wires}}{\text{surface area of ideal anode sphere}} , \tag{2.16}$$

$$G_{t,c} = \frac{\text{surface area of metal wires}}{\text{surface area of ideal cathode sphere}} . \tag{2.17}$$

These transparencies can be approximated using the wire diameter d_w, the number of great circle wire components N_s used in the construction of the anode or cathode, the cathode radius R_c , the anode radius R_a , the number of anode latitude wires N_L, and the anode latitude circle radius R_L as

$$G_{t,a} \cong \frac{N_s 2\pi R_a d_w + N_L 2\pi R_L d_w}{4\pi R_a^2} , \tag{2.18}$$

$$G_{t,c} \cong \frac{N_s 2\pi R_c d_w}{4\pi R_c^2} . \tag{2.19}$$

In the PIC simulation, when the macroparticle is found to have moved across the anode or cathode, a random number is generated between 0 and 1. If the macroparticle crossed the anode, the random number is assessed and if it is greater than $G_{t,a}$, the particle is treated as if it collided with the anode grid, and the particle is deleted from the problem domain. If the random number is less than the geometric transparency, then the macroparticle is unchanged and is considered to have passed through the anode without a collision. This same logic is applied to crossing the cathode, with $G_{t,c}$ replacing $G_{t,a}$.

The macroparticle source consists of a ring of thickness d_s near the anode ring at a radius R_s from the center of the system shown in Figure 2.3.

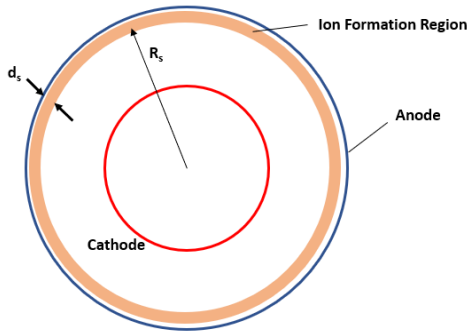


Figure 2.3 The macroparticle generation region is taken to be a band of width d_s near the anode. This region is where the radial electric field ionizes the background deuterium gas. The macroparticle ions are formed at a rate of *I* particles per timestep and are initialized randomly in the orange band. A Maxwellian distribution of particle velocities is sampled at room temperature to initialize the particle x and y velocities.

This region, shown in orange in Figure 2.3, is the region in which ionization of the background deuterium gas occurs in the model. The macroparticle ions are formed at a rate of *I* macroparticles per timestep, and they are initialized at a random position within the orange band in Figure 2.3. These particles are initialized with a thermal velocity generated randomly by sampling a Maxwellian distribution of x and y velocities at room temperature. The ion source flux, in ions/second, was assumed to be a simple linear function of the cathode voltage:

$$I = 1.6 \times 10^{21} \frac{|\phi_{cathode}|}{100{,}000} \tag{2.20}$$

Here $\phi_{cathode}$ is the cathode voltage, and 1.6×10^{21} was used as an adequate ion source flux because when used in the PIC simulation it produced a stable ion density at 100,000 V that produced maximum ion densities on the same order of magnitude as experiment [1]. This is a rough approximation of true gas discharge physics, and will be modified in future research to more accurately model the discharge behavior of deuterium gas in an external electric field.

2.1.4 Cross Section Implementation

The macroscopic D-D fusion cross section $\sum_f$ was introduced in Section 1.2.1 and depends on the ion density n_{ion}, and the ion microscopic cross section, $\sigma_f(\mathbf{v})$ according to Equation 1.5. n_{ion} is a function of position in the IEC device. This makes the macroscopic cross section $\sum_f$ dependent on both the position and velocity of the interacting ions. Thus, as the particles are moved through the computational domain,

they have a probability of undergoing a fusion reaction depending on the distance they travel, the local ion density, and the instantaneous velocity at which the ions are traveling given by Equation 2.15. A fusion event is considered an ion loss mechanism in the computational model, as discussed in Section 2.1.3.4.

In this study, the energy dependence of the microscopic fusion cross section for a single ion $\sigma_f(E)$ is modeled using the NRL 5-parameter fitting formula [6] given by the following equation:

$$\sigma(E) = \frac{1}{E\left(e^{\frac{A_1}{\sqrt{E}}} - 1\right)}\left[A_5 + \frac{A_2}{(A_4 - A_3E)^2 + 1}\right]. \tag{2.21}$$

Here A_1, A_2, A_3, A_4, and A_5 are experimentally determined parameters that take on unique values for each nuclear reaction, and are listed below in Table 2.1. These values require an energy input in keV and will output a cross section in barns. Because D-D fusion can yield two different sets of products with a 50% branching ratio (Equations 1.2 and 1.3), there is a different set of five parameters characterizing these two reactions for a total of ten parameters. These are listed in Table 2.1. The total microscopic cross section for a fusion reaction to occur is simply the sum of these two cross sections given by

$$\sigma_{Total}(E) = \sigma_1(E) + \sigma_2(E)\,. \tag{2.22}$$

Expressing Equation 2.21 in terms of the ten fitting parameters and writing in terms of velocity by replacing E with $\frac{1}{2}mv^2$ yields the desired form of the total deuterium fusion microscopic cross section in terms of velocity dependence in the laboratory frame:

$$\sigma_{Total}(v) = \frac{2\left[A_5 + \frac{A_2}{\left(A_4 - A_3\frac{1}{2}mv^2\right)^2 + 1}\right]}{mv^2\left(e^{\frac{A_1}{\sqrt{\frac{1}{2}mv^2}}} - 1\right)} + \frac{2\left[A_{10} + \frac{A_7}{\left(A_9 - A_8\frac{1}{2}mv^2\right)^2 + 1}\right]}{mv^2\left(e^{\frac{A_6}{\sqrt{\frac{1}{2}mv^2}}} - 1\right)}. \tag{2.23}$$

Table 2.1

Parameter	Value
A_1	46.097
A_2	372
A_3	4.36×10^{-4}
A_4	1.220
A_5	0
A_6	47.88
A_7	482
A_8	3.08×10^{-4}
A_9	1.177
A_{10}	0

This function is plotted along with experimental data from the ENDF/B VII.0 dataset from [6] and is shown below in Figure 2.4. The 5-parameter fit closely matches the experimental data below energies of roughly 200 keV. The energies encountered in the device proposed in this study are expected to be on the order of 20 keV, well within the 5-parameter fit's range of accuracy.

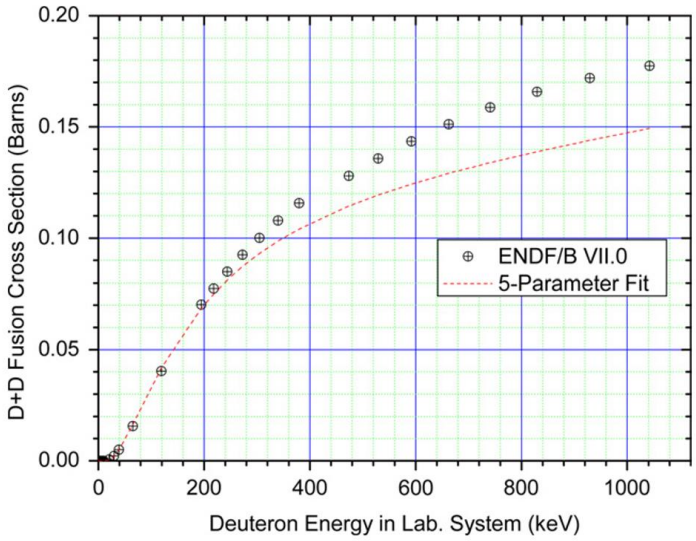


Figure 2.4 Deuterium fusion microscopic cross section data from ENDF/B VII.0 dataset (black circles) and the 5-parameter fit (dotted line). The 5-parameter fit models the data well for energies less then roughly 200 keV. Used with permission from Ref [6].

It is assumed all w_s ions within a macroparticle have identical energy and position, so their cross sections are the same.

2.2 System Overview

The hypothetical physical system proposed in this study that is being modeled by the PIC computational model consists of a cylindrical aluminum vacuum chamber, three ceramic stalks used to support the anode and cathode wire grids, the anode and cathode wire spheres, and electrical and vacuum systems. Figure 2.5 shows the assembled chamber support structure and a section cut view of the main chamber in a qualitative manner. Dimensions are shown in the supsiquent engineering drawings (Figure 2.6 a. and b.).

a) *b)*

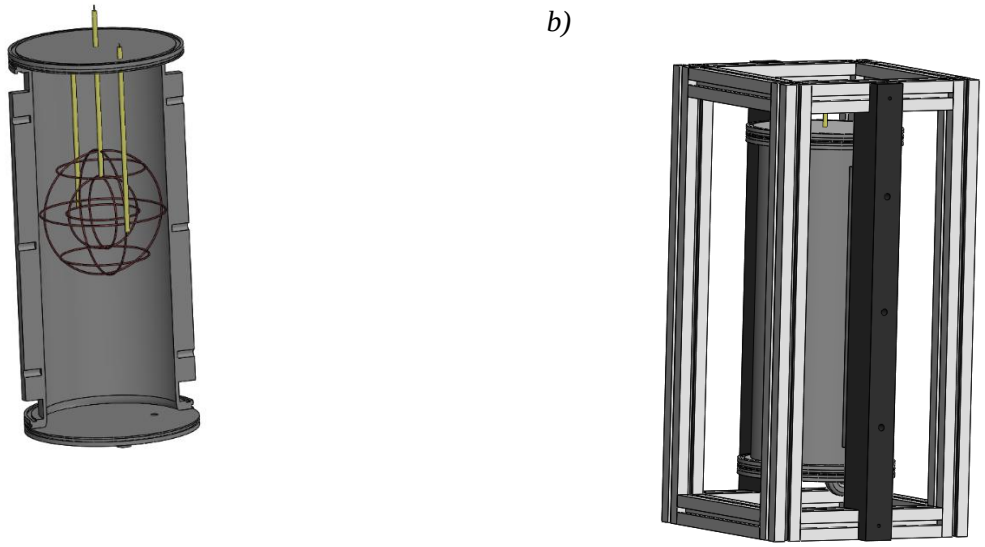


Figure 2.5 a. Section cut view of vacuum chamber and IEC grid assembly with high voltage ceramic stalk support structure. Figure 2.5 b. Assembled vacuum chamber, IEC grid, and surrounding support structure.

a) *b)*

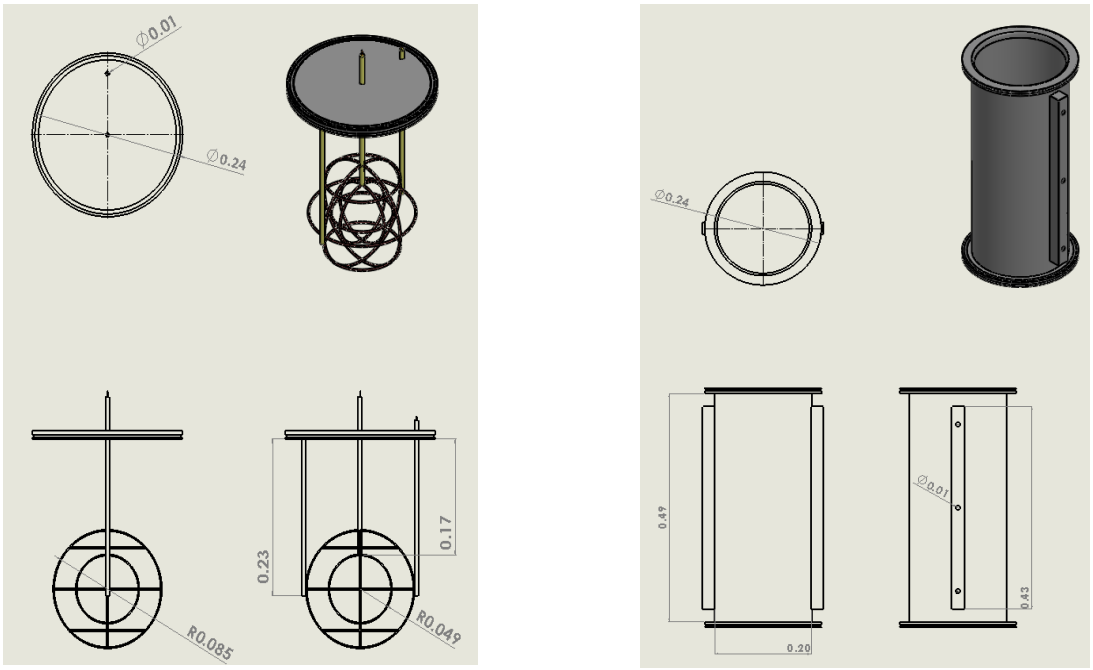


Figure 2.6 a. Engineering drawing for top flange and IEC anode and cathode wire meshes. Dimensions are in meters. The anode and cathode stalk lengths are shown, and the anode and cathode radii are also shown. Figure 2.6 b. Engineering drawing for the vacuum chamber, in meters.

The wall voltage is stepped up to 20 kV DC and is applied to the cathode grid (the smaller of the two wire spheres). The anode (the larger sphere) is grounded. Ceramic tubes (shown in yellow in Figure 2.5 a.) are used as high voltage stalk to protect the incoming high voltage wires from plasma bombardment and provide structural support. The vacuum chamber should be brought down to a steady state pressure of roughly 7 Pa. The measurement of 2.45 MeV fusion neutrons should be carried out by a scintillation neutron detector surrounded by a Bonner sphere neutron moderator, which is discussed in detail in Section 2.4. A general diagram of the entire system is seen below in Figure 2.7.

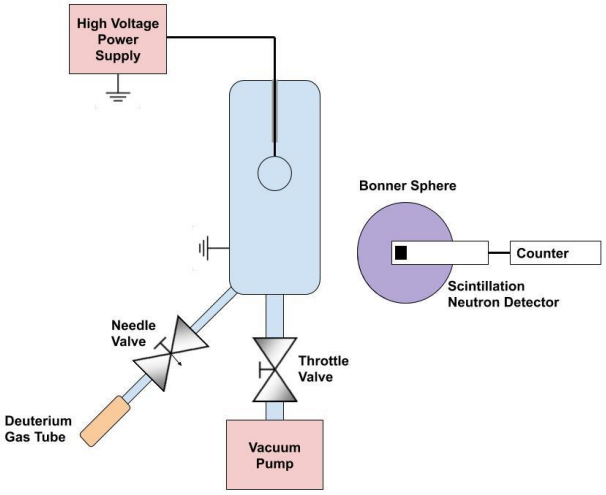


Figure 2.7 Schematic of proposed system setup. The high voltage power supply enters the vacuum chamber and imparts a 15 – 30 kV potential onto the cathode. Deuterium is fed into the system through the needle valve, and the vacuum is maintained by a rotary vane vacuum pump.

2.3 Neutron Radiation Safety Risk

Neutron radiation has harmful effects on biologic tissue if proper safety issues are not considered [7]. Because this study requires neutron radiation counts to verify the occurrence of nuclear fusion, safety precautions were used when designing the IEC to ensure safe operations.

In order to design safe experimental practices, it is necessary to predict the magnitude of neutron flux φ at a distance D from the center of the IEC felt by experimenters near the device. To estimate the expected total neutron rate of production in the D-D (deuterium-deuterium) IEC device, measurements made of the neutron production rate versus the grid voltage and metered current for D-D fusion are shown in Figures 2.8 and 2.9 [8]. These figures illustrate that the neutron production rate inside the IEC is strongly dependent on the applied grid voltage and weakly dependent on the metered current to the system.

In this thesis, the proposed IEC device operates at a maximum of 30 kV and 30 mA (the simulation will be taken up to 200 kV, however). These levels are treated as ceiling limits to supplied voltage and current.

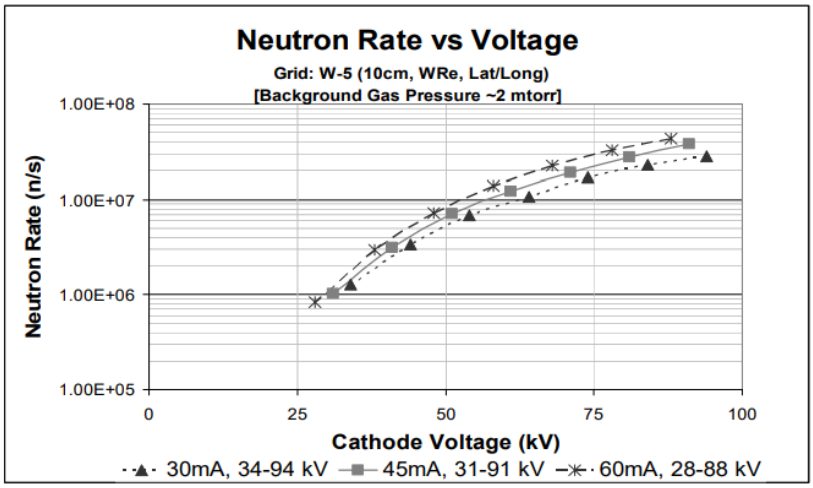


Figure 2.8 Total neutron production rate vs voltage for a D-D 10 cm diameter cathode IEC [8]. The neutron production rate is a strong function of grid voltage. Figure adapted from Ref [8]. Reproduced with permission.

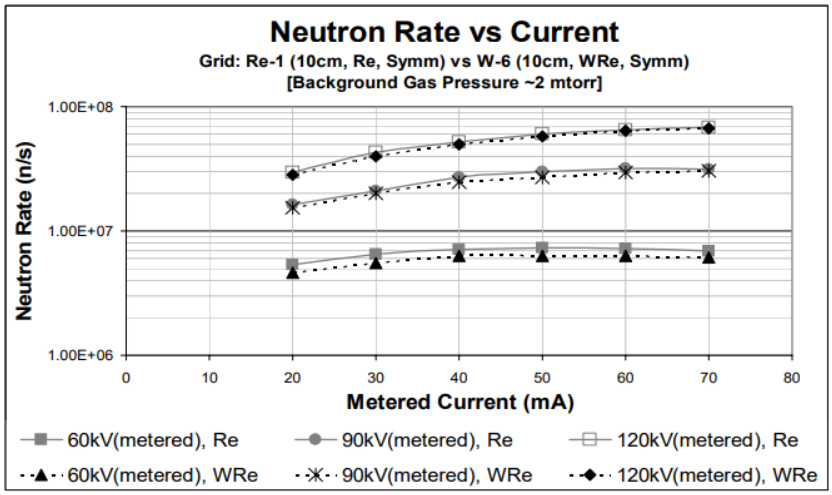


Figure 2.9 Total neutron production rate vs current for a D-D 10 cm diameter cathode IEC [8]. The neutron production rate is a weak function of current. Figure adapted from Ref [8]. Reproduced with permission.

At these maximum values, the neutron production rate from Figure 2.8 is on the order of 8.00×10^5 neutrons per second. The 8.00×10^5 production rate at 30 kV and 30 mA can be treated as a worst-case scenario for the proposed IEC in this project.

Treating this maximum rate as a point source at the center of the cathode, the maximum neutron flux that could be felt by a researcher would be at 10 cm if one were to place a hand on the outside of the vacuum vessel. The flux at this distance is given by Equation 2.24 assuming no attenuation (worst case):

$$\varphi = \frac{S}{4\pi D^2} \tag{2.24}$$

Here φ is the neutron flux in neutrons per cm^2 per second, D is the radial distance from the center of the cathode wire grid, and S is the source strength, which at worst is 8.00×10^5 n/s. The felt flux at 10 cm is:

$$\varphi_{worst\ case} = 636.62 \frac{n}{cm^2 s}.$$

For safety, no person will ever be this close. The operator will be at a minimum safe distance of 5 meters from the machine, which yields:

$$\varphi_{worst\ case\ at\ safe\ distance} = 0.255 \frac{n}{cm^2 s}.$$

This value is well within acceptable radiation limits [7] for the expected total operation time of roughly 16 minutes.

2.4 Bonner Sphere High Energy Neutron Detection

The 2.45 MeV neutrons produced by D-D fusion have too high an energy to be detected by available neutron detectors. This can be changed by slowing down the neutrons to more modest (and detectable) energies by using a neutron moderator, which is a material that exchanges energy with neutrons through collisions. The neutron detection method used is a LiI(Eu) scintillation detector surrounded by a Bonner sphere [9] made from a polyethylene spherical shell shown in Figure 2.7. Polyethylene is a neutron moderator and can reduce the neutron energy enough for detection. Bonner Spheres are reportedly able to detect neutrons with energies as high as 20 MeV as shown in Figure 2.10, well-equipped to handle the 2.45 MeV neutrons from D-D fusion occurring in the IEC.

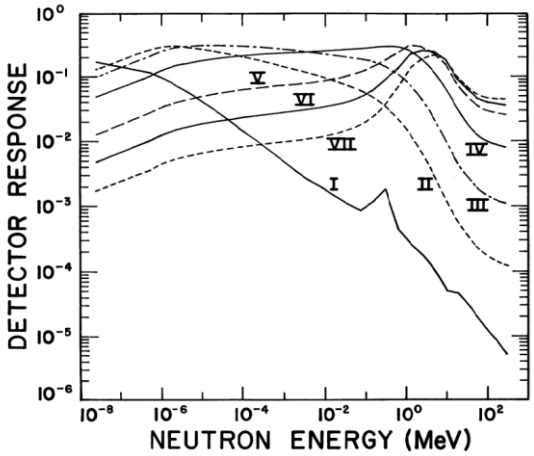


Figure 2.10 Detector response as a function of neutron energy [9]. Roman numerals correspond to Bonner Spheres with increasing diameter: 2, 3, 5, 8, 10, and 12-inch diameter. The 2.45 MeV neutrons produced from D-D fusion in this study are best suited for the 5-inch diameter Bonner sphere. Figure reproduced with permission from [9].

2.5 High Voltage Power Supply

High voltage is supplied by the wall outlet and will be stepped up and converted to DC at a maximum of --30 kV (Figure 2.11). The current will be on the order of 10 mA.

Safety precautions to mitigate risk from the high voltage power supply to the central grid include a kill switch and fuse to cut power, and a standard operating procedure (SOP) regarding discharging the device.

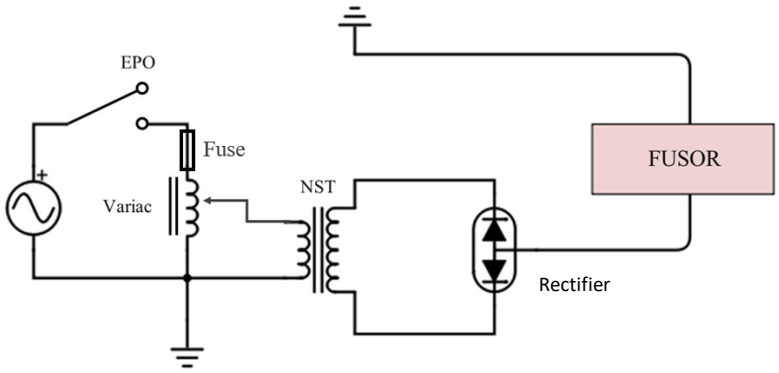


Figure 2.11 The general schematic of the proposed electrical system, which includes an emergency power off (EPO) switch and a fuse. The Variac and NST step up the wall voltage. The high voltage rectifier converts AC to DC [10].

The Variac and the NST step up the wall voltage from low voltage, high current AC to high voltage, low current AC. The high voltage rectifier is constructed using two high voltage diodes. This component provides the final conversion from high voltage, low current AC to high voltage, low current DC.

2.6 High Voltage Feedthrough

The high voltage feedthrough is constructed using a tube of non-porous high alumina ceramic. Its purpose is to provide structural support to the cathode grid as well as serve as the electrical connection between the cathode central grid and the high voltage power supply. It must be built from a material that is rigid, insolating, and has a high melting point [1]. Thus, ceramics offer the optimum choice to serve these purposes.

2.7 Vacuum Chamber

The vacuum chamber used in the proposed experiment is shown in Figure 2.6 b. It is made from aluminum which is grounded during operation of the device. The vacuum chamber was purchased used and initially had only one flange and was filled with an undesirable black paint finish on the interior. This finish was removed to avoid sputtering paint into the plasma. Two new flanges were purchased. One was a blank, which was machined to include the input ports for the feedthrough and vacuum pump. A hole was bored through the second flange and a viewport was tack-welded to it to monitor the plasma behavior with a camera. This viewport aims away from the device operators during operation due to possible X-ray leakage.

2.8 Anode and Cathode Construction

The cathode grid for the proposed device was constructed out of tungsten 20-gauge wire. This material was chosen due to its high melting point: the cathode grid is exposed to much higher heat fluxes then the anode grid, which is built from stainless steel 20-gauge wire. The fabrication process involves wrapping wire around Styrofoam balls of the correct radius in the desired geometry and spot-welded at intersections. The Styrofoam is then dissolved away using acetone, leaving the wire grid behind.

Chapter 3 – Results and Discussion

Tunable parameters such as the electron temperature, electron density, and ion source flux needed to be determined before the simulations were run. An electron temperature of 600 eV and an electron background density (n_o in Equation 2.5) of $1x10^{16}$ electrons/m^3 were found to be appropriate values for these two tunable simulation variables. These values were selected from [11], who studied the electron temperature and density profiles for IECs. The values used corresponded to lower grid potentials than the ones used in this study, however, and could be a source of error. The ion source flux was implemented using Equation 2.20, which assumes the ion source strength is linearly related to the cathode voltage. The slope of the linear ion source flux was chosen to yield 1.6×10^{21} ions/s at a cathode voltage of -100 kV, which yielded stable ion densities that were on the same order of magnitude as experiment [1].

The extremely fine timestep of 0.1 ns lead to long simulation times while waiting for fusion reactions to occur. Figure 2.8 suggests that at a grid voltage of 50 kV, the expected fusion rate should be on the order of $1x10^7$ fusions/s, which when multiplied by the timestep of 0.1 ns yields 0.001 fusion reactions per timestep. This means that on average 1000 time steps will occur between each fusion reaction. It took the computer on which the simulations were run approximately 30 seconds to finish one timestep, so it took roughly 8.3 hours to see one fusion reaction occur in the simulation. To combat this long wait time, the probability of fusion was artificially enhanced by a factor of 1×10^4 in order to see more frequent fusions in a shorter amount of real time. This factor brings the average real time between fusions to ~3 seconds. This artificial factor was then divided out of the resulting average fusion rate over the 80 ns simulation time to yield the number of fusions per 80 ns. These rates were used as the average fusion rate at a given cathode voltage, in fusions per second. The cathode grid voltage was varied, and the average fusion rate at that cathode voltage was recorded. A geometric transparency (Equations 2.18, 2.19) of 0.95 was used for both the cathode and anode. A specific weight w_s of 6×10^8 ions per macroparticle was used.

3.1 Ion Densities and Fusion Rates

To visualize the potential field on the computational domain, the PIC algorithm was run with no particles present. This was used as a base case to ensure that the potential solver was working properly, as the solution can be readily confirmed by an analytical calculation. The results of the no-particle base case potential solution can be seen below in Figure 3.1. As expected, the potential is constant within the central cathode, and tapers off linearly in the radial direction toward the anode ring. A cell size of $dx = dy = 4$

mm was used instead of the system Debye length $\lambda_D = 1.821$ mm. This compromise was made to avoid extremely long simulation times. Physically, using a grid spacing larger than λ_D is acceptable because a plasma exhibits quasineutrality at all lengths above λ_D. Thus, the larger grid spacing ensures that particles within any particular cell will experience an electric field only from the corners of the cell it currently occupies.

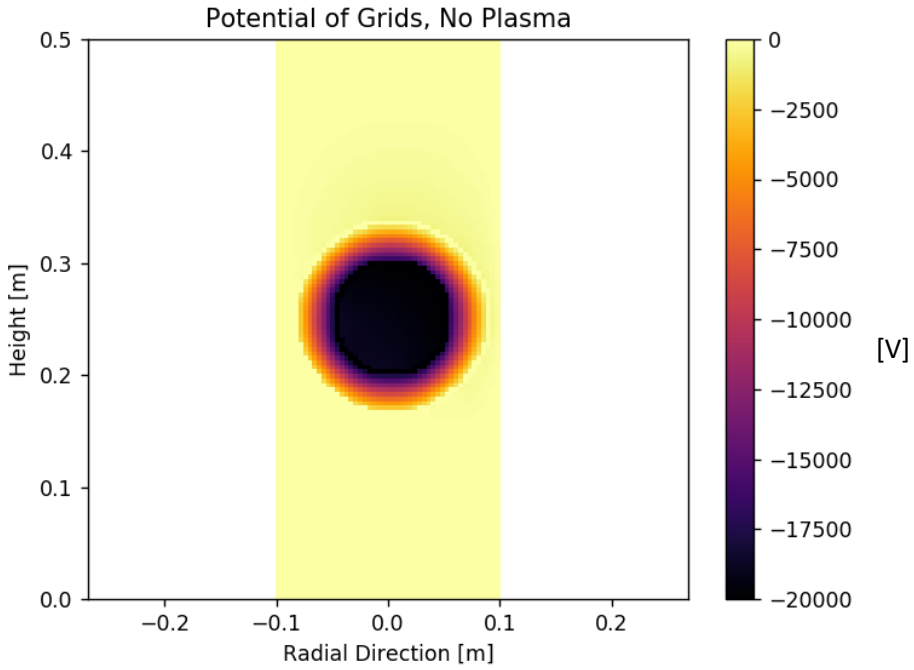


Figure 3.1 Contour plot of system potential outputted by the PIC algorithm with no particles present at a cathode voltage of -20 kV. This builds confidence in the potential solver because this solution can be readily verified by an analytical treatment of concentric equipotential rings. Note the individual cells can be seen at this scale.

After the no-plasma base case, a series of simulations were run while varying the cathode grid potential, keeping all other parameters held constant (except for the ion source flux, Equation 2.20). The time step was set to $1\text{x}10^{-10}$ s in order to capture the extremely high velocities seen in this device, and 800 time steps were simulated resulting in a total simulation time of 80 ns. The cathode potential was varied in 10 kV increments from 10 kV to 200 kV, which is higher than what is expected in the proposed physical device (maximum of 20 kV), but similar to other IEC experiments [1]. Density profiles, potential profiles, and fusion rates along with the time and position of each fusion reaction were recorded in csv files. Below in Figure 3.2, the ion density and potential distribution within the simulated IEC be seen for a cathode grid voltage of -70 kV.

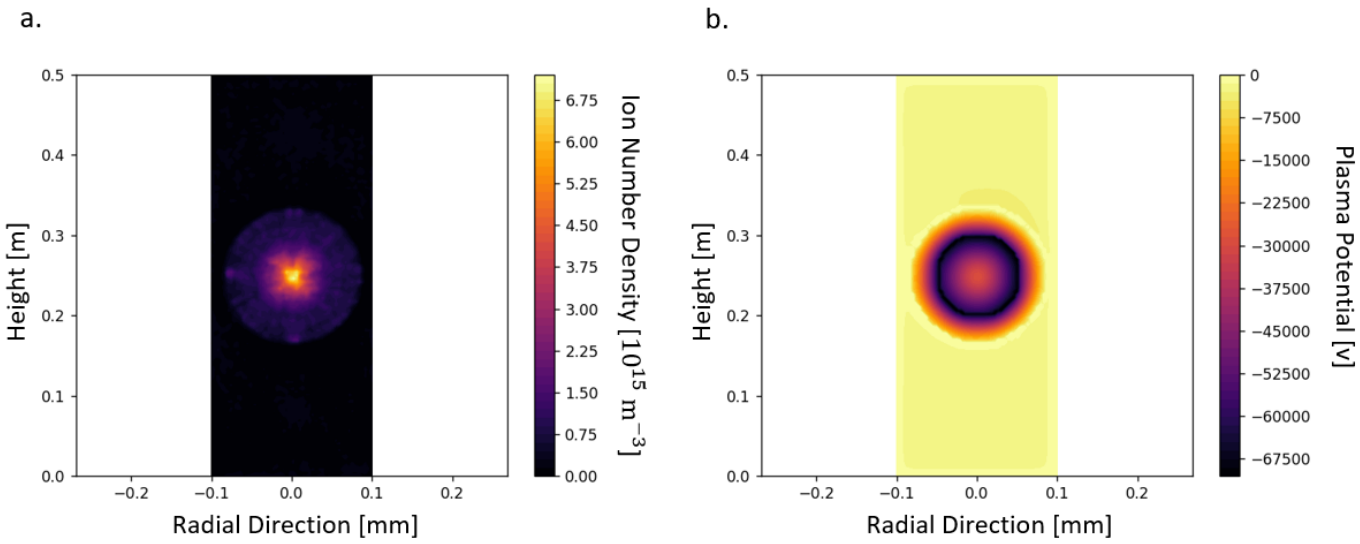


Figure 3.2 a) The plasma ion density is shown above after 80 ns of motion through the potential created by a cathode at -70 kV. A central dense region can be seen which builds confidence that the simulation is capturing key aspects of IEC behavior. b) Potential distribution, demonstrating the effect of the positive ions on the potential.

The ion number density field from the initial PIC run shows promising results. The central core of the IEC in Figure 3.2 a. shows a density spike, matching the expected behavior of the IEC device (Figure 1.2 b). The peak density can be seen to taper off from the core region, and then increase in magnitude slightly as the source radius is reached. This behavior is expected: the slow-moving particles near the source radius build up density there, but as they pick up velocity and fall through the core, they converge on the center of the device and build up much higher ion densities. The potential distribution (Figure 3.2 b.) demonstrates that as the positively charged deuterium ions build up in the core, they shield the negative cathode potential. Thus, a bump of increased potential can be seen at the center of the device where the largest ion density occurs. The effect can be easily seen when comparing Figure 3.1 with Figure 3.2 b. For the complete set of 20 simulation density and potential plots, see Appendix A.

A radial cut of the ion density through the core was performed to visualize the ion density as it varies in the radial direction. Radial density distributions along the short axis of the device are shown for every simulated grid voltage below in Figure 3.3 a.

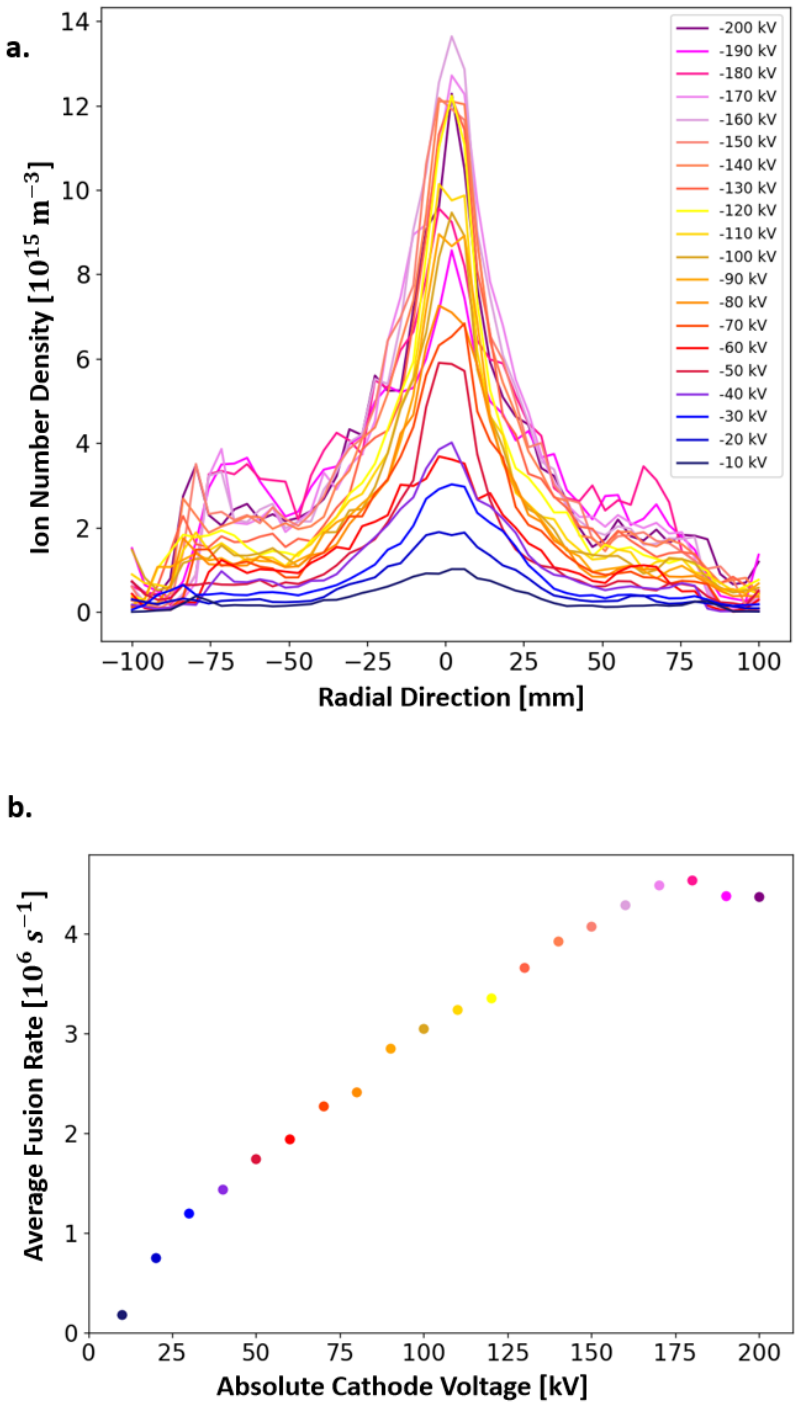


Figure 3.3 a. The ion density along a line through the center of the IEC shows how the ion density varies with radial distance in the device. This density profile is plotted for different cathode grid voltages, demonstrating the density build up in the core as the grid voltage is increased. Figure 3.3 b. demonstrates the fusion rate's dependence on grid voltage. These plots were taken from simulations run over an 80 ns duration of simulated time. Each timestep was 0.1 ns, for a total of 800 timesteps. The fusion rate reported is the average fusion rate seen over the 80 ns.

Figure 3.3 b. shows the corresponding fusion rate for each simulated cathode potential. The fusion response to cathode grid voltage increases nearly linearly from -10 kV to -180 kV, and then begins to drop off as the cathode voltage is further increased. This drop-off behavior is likely due to the constant electron temperature and density that was used throughout the simulations, along with the assumption that the ion source flux due to the electric field strength varies linearly with the cathode-to-anode potential difference. The electron parameters used from [11] were for low-voltage (~4 kV) voltage differences, so considerable error may come from pushing the magnitude of the cathode voltage up in the simulations undertaken here past where the assumed background electron density is reasonable.

For the lower voltages which were simulated here, the background electron density used from [11] is a better approximation for the true electron background density in a devise at the given cathode voltage. By comparing the simulated fusion rates at the lower voltages with experimental neutron production rates made by [12] in a deuterium-deuterium IEC, the simulation can be seen to be capturing a similar order of magnitude as [12], along with a correct increasing fusion rate with cathode voltage. However, the concavity of the simulated results does not match the concavity of either of the measured results in Figure 3.4 a. This again is likely due to the constant background electron density, which is only strictly valid at low cathode voltages.

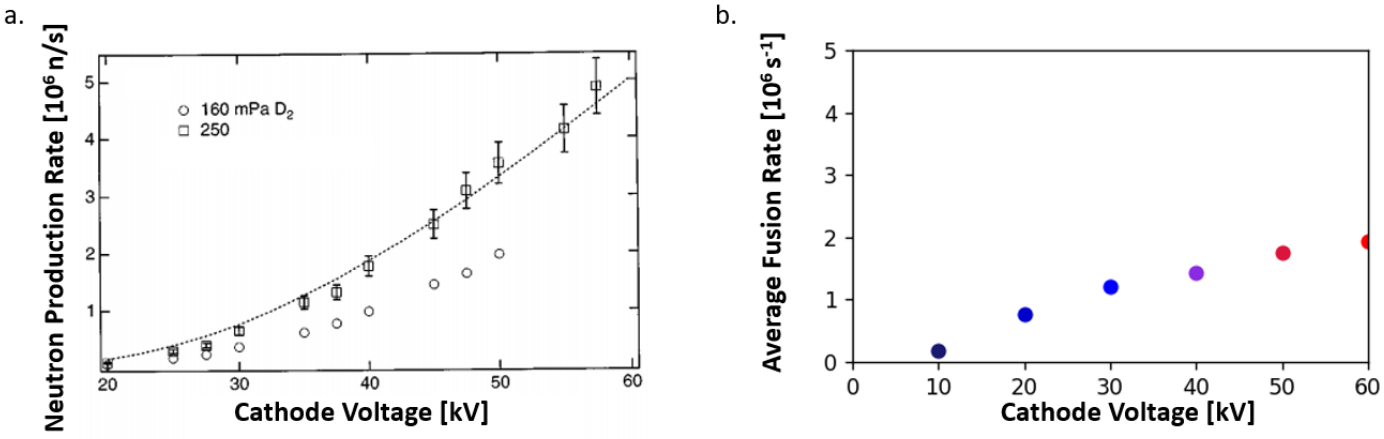


Figure 3.4 a. Total neutron production rate from measurements on an IEC device of comparable size to the device modeled in this thesis, adapted with permission from [12]. Note that both datasets seen in a. are for D-D fusion, but operating at different pressures. b. Sample of fusion rates from Figure 3.3 b. in a similar voltage range to 3.4 a. Note that fusion rate and neutron production rate are not strictly comparable owing to the branching ratio for the D-D fusion reaction (Equations 1.2, 1.3). The 50% branching ratio means 3.4 b. is scaled up by a factor of 2 over the neutron production rate in the IEC simulated in this thesis.

An assumed electron background density that is too low would not adequately shield the buildup of positive space charge in the center of the IEC device, the net effect being a stronger repulsion to ions seeking to coalesce at the core of the device. This ion density drop seen in the higher voltage regime due to the low electron density would limit the amount of fusion that could occur. Figure 3.3 a. shows that for the highest magnitude of cathode voltages simulated (-180, -190, and -200 kV), the peak densities are lower than for cathode voltages of -160 and -170 kV, indicating that the ion repulsion resulting from the low electron density is effecting the simulation's validity at high cathode voltages. This is confirmed by the drop seen in the fusion rate in Figure 3.3 b at -180, -190, and -200 kV.

3.2 IEC Fusion Rate Startup Behavior

The time behavior of IEC devices is not extensively covered in the literature due to the small timescales involved in ion cloud formation and fast transition to steady-state fusion operation. Due to the PIC simulation's small timestep of 0.1 ns, the fusion reaction's response to time at steady cathode voltages could be seen in detail. In order to present the fusion counts seen in the simulation as a fusion rate, the 80 ns computational time was divided into 12 bins of size 6.667 ns. The number of fusion events which occurred in each bin were counted, and the fusion rate was calculated by dividing the number of counts seen in each bin by the bin size, and the factor of 1×10^4 used to enhance the probability of fusion. The resulting fusion rate versus time behavior can be seen below in Figure 3.5.

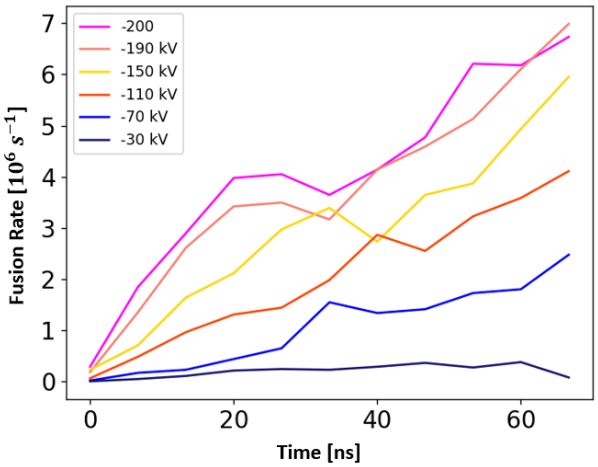


Figure 3.5. Start-up response of an IEC as a function of cathode grid voltage. The fusion rate was calculated from fusion counts recorded in the simulations by collecting fusion events into 12 bins discretized in time. Each bin counted the total number of fusions occurring within 6.667 ns ranges of time. The number of fusions within each bin was divided by the bin time range, 6.667 ns, to yield the average fusion rate over that bin.

The fusion rates for all but the -10 kV case can be seen to increase with time as the density in the IEC builds up. The fusion rates are on average higher at nearly all times for higher cathode voltages, with the exception of high magnitude cathode voltages. This drop in the fusion rate at high magnitude cathode voltages is likely explained by the ion repulsion caused by the low background electron density used in the simulations.

Chapter 4 – Conclusion

Inertial electrostatic confinement (IEC) fusion offers a low-cost and practical means to study nuclear fusion without needing large and expensive magnetic confinement devices. Researchers can use findings from fusion studies on IEC's to supplement and guide larger fusion devices such as the next generation of tokamaks. In addition to energy, fusion also offers an on-demand way to produce high neutron fluxes for neutron activation analysis. Studying the rate at which fusion occurs in an IEC can help match IEC grid voltages to required neutron flux for neutron activation analysis studies.

The primary objective in this thesis was to create a particle-in-cell plasma simulation to study how the fusion rate and ion density distribution in an IEC is affected by the cathode grid voltage. A simple numerical fusion collision model based on the 5-parameter fit [6] for fusion cross sections was proposed (Equation 2.15) and implemented as an addition to the traditional particle-in-cell computational plasma model, which usually excludes any particle collisions [5]. This enables the study of fusion reactions with the particle-in-cell method.

It was found that the IEC ion density peak raises with increasing cathode voltage (Figure 3.3 a.), along with a wider region of heightened density surrounding the core. Densities seen in the simulation were on the order of 6×10^{16} deuterium ions per m^3, similar to those published in Ref [1]. The fusion rate increased nearly linearly with increasing grid voltage. The highest fusion rate seen (Figure 3.3 b.) was 4.54×10^6 fusions per second for the -180 kV cathode voltage. The densities and fusion rates in the high magnitude cathode voltage regime were lower than expected, likely due to the low background electron density used in the simulations.

Due to the small simulation timestep, the time response of the fusion reaction rate at different cathode voltages could be studied. IEC start-up behavior is seldom found in the literature; thus Figure 3.4 offers insight into a seldom-studied aspect of IEC behavior.

4.1 Future Research

This study was conducted using an ion source flux that scaled linearly with cathode grid voltage. Future researchers applying the particle-in-cell IEC code developed here should seek out a better expression for deuterium ionization in an external electric field. In addition, the Maxwellian treatment of electrons

should be treated with caution, as the IEC device is generally a non-equilibrium device [1]. The electron temperature and density used was for a much smaller cathode voltage than those simulated here. Along with the ion source, future researchers should implement a more physically realistic expression for the background electron density. The PIC code developed here could also be extended from 2D to 3D to simulate the IEC plasma dynamics more realistically.

The hardware purchased for constructing the 20 kV IEC proposed here is currently housed in the Radiation Center at Oregon State University. All materials and equipment are present, with the exception of neutron detection equipment, which are available from the Nuclear Engineering Department. When the restrictions on lab work due to COVID-19 are lifted, the completion and testing of the physical IEC device modeled in the simulations will be conducted to compare the computational fusion rate to that measured on the physical machine.

Acknowledgments

This project was funded by a NASA Undergraduate Research Fellowship Award, so I would to thank NASA and the Oregon Space Grant Consortium for offering career building research experiences to undergraduate students. I would like to thank Dr. Brian Woods for his outstanding and patient leadership during my two years of working on this IEC project, and for him sparking my passion for engineering applications for fluids and plasmas through his fluid mechanics and plasma physics courses. I would like to thank Dr. David Roundy for substantially speeding up the PIC algorithm by suggesting an iterative approach to solving for the system potential rather than with matrix inversion methods. I would like to thank Lubos Brieda, whose blog on the PIC method and MATLAB examples were used as the foundation of the PIC python implementation used in this thesis. I would like to thank Dr. Janet Tate and Dr. Ethan Minot for a year of support in the writing process of the thesis. Finally, I would like to thank Kaylin Gopal, who worked with me as an Oregon State University undergraduate to undertake this IEC project and who participated in countless hours of brainstorming.

References

1. Krupakar Murali et al. “Inertial Electrostatic Confinement (IEC) Fusion: Fundamentals and Applications”. Springer, 2016.

2. Shultis, J. Kenneth et al. “Fundamentals of Nuclear Science and Engineering”. CRC Press, 2017.

3. Bonomo, Richard. “Inertial Electrostatic Confinement Fusion.” Inertial Electrostatic Confinement Project - University of Wisconsin - Madison, iec.neep.wisc.edu/.

4. Inan, Umran S., and Marek Gołkowski. *Principles of Plasma Physics for Engineers and Scientists*. Cambridge University Press, 2011

5. Brieda, Lubos. “The Electrostatic Particle In Cell (ES-PIC) Method.” PICC Blog RSS, https://www.particleincell.com/2010/es-pic-method/.

6. Li, Xing et al. “A New Simple Formula for Fusion Cross-Sections of Light Nuclei” *Nuclear Fusion 48(12):125003*. November 2008

7. *Biological Effects of Radiation.* USNRC Technical Training Center.

8. Wehmeyer, A.l., et al. “Optimizing Neutron Production Rates from D-D Fusion in an Inertial Electrostatic Confinement Device.” *Fusion Science and Technology*, vol. 47, no. 4, 2005, pp. 1260–1264., doi:10.13182/fst05-a861.

9. Awschalom, Miguel. “Applications of Bonner Sphere Detectors in Neutron Field Dosimetry.” Fermilab Neutron Therapy Facility, September 1983

10. Spranger, Dan. “Learn How to Build a Nuclear Fusor” Maker Magazine. https://makezine.com/projects/make-36-boards/nuclear-fusor

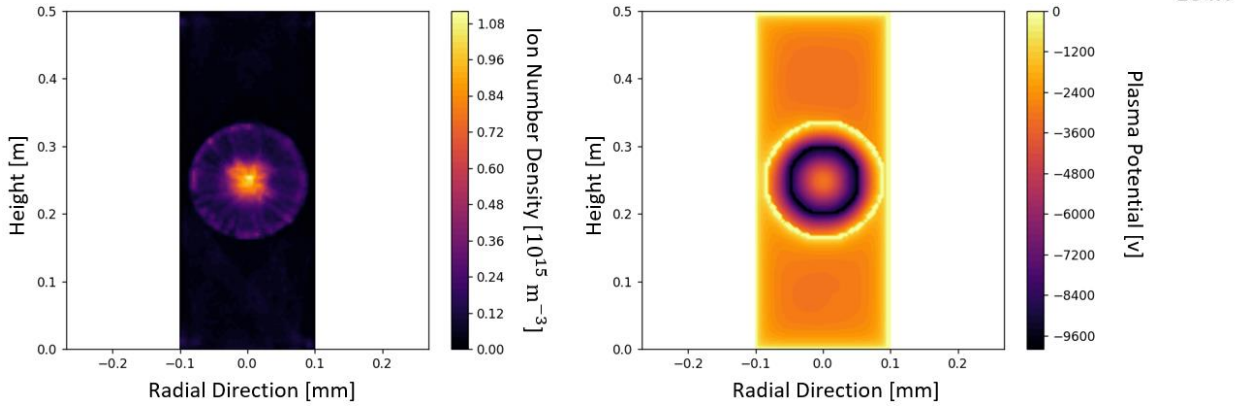
11. Meyer, Ryan. “Inertial Electrostatic Confinement : Theoretical and Experimental Studies of Spherical Devices.” Dec. 2007, doi:10.32469/10355/4736

12. Thorson, T.a, et al. “Fusion Reactivity Characterization of a Spherically Convergent Ion Focus.” Nuclear Fusion, vol. 38, no. 4, 1998, pp. 495–507., doi:10.1088/0029-5515/38/4/302.

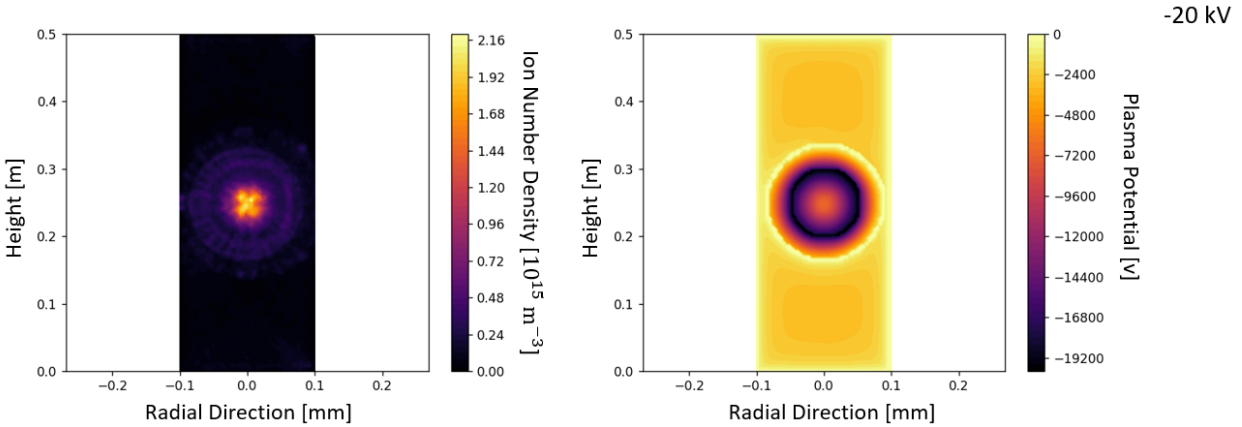
Appendix A: Density and Potential Profiles

Shown here are the deuterium ion number density and system voltage plots for each of the 20 simulations that were run to produce Figures 3.2, 3.3 a and b, 3.4 b, and 3.5. The voltage in the upper right of each figure is the cathode voltage used for the respective density and system voltage plots. The cathode voltage magnitude is increased with each figure.

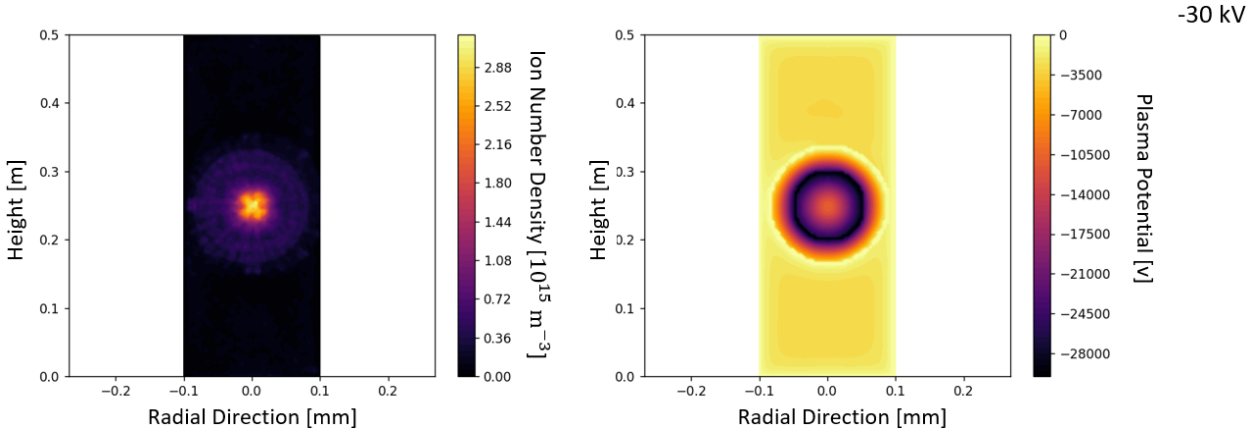
-10 kV



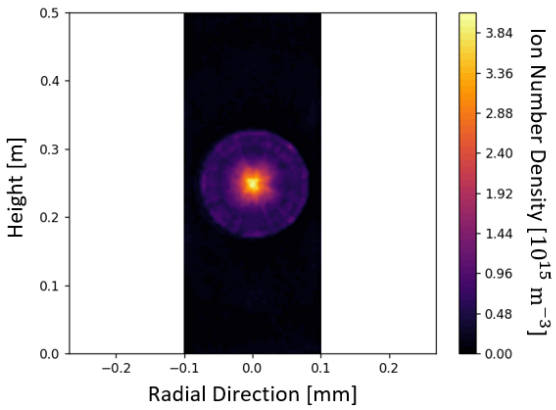
-20 kV

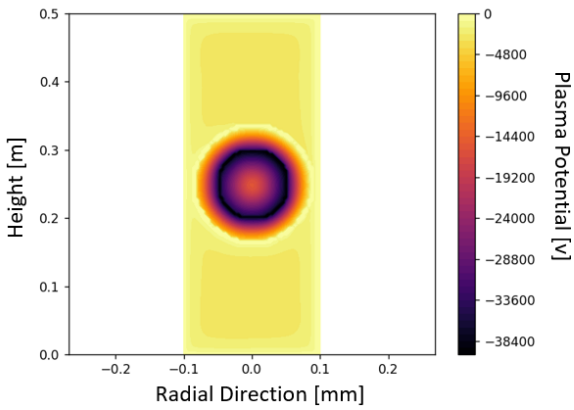


-30 kV

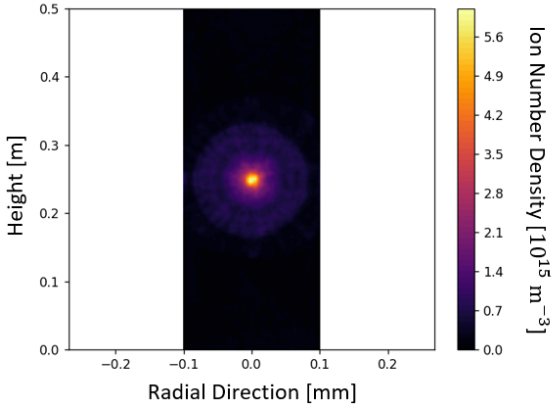


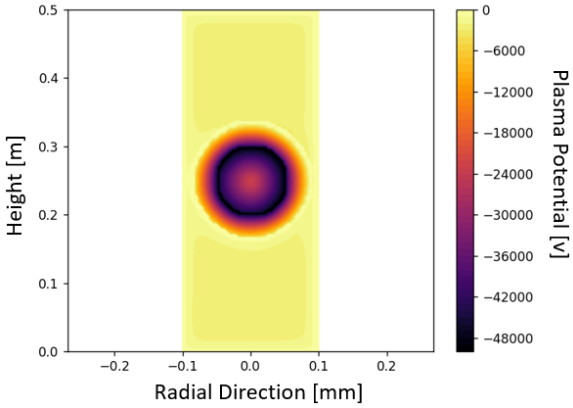
-40 kV



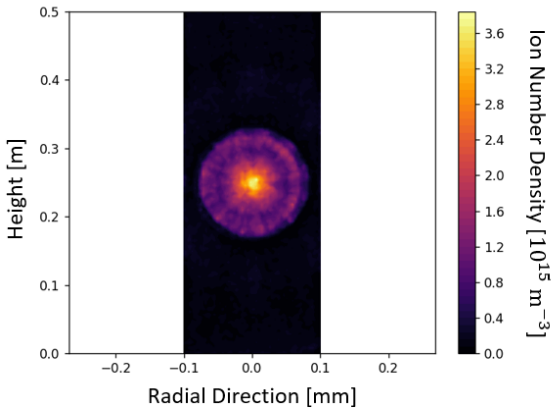


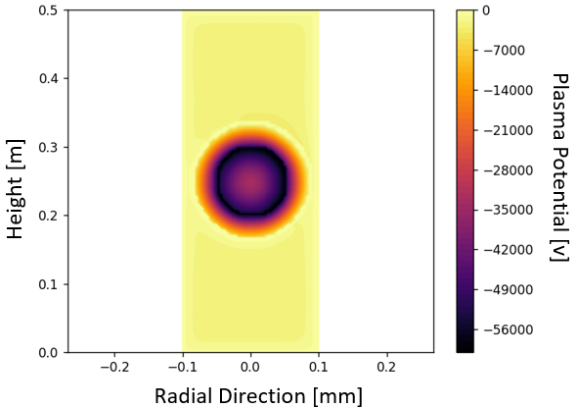
-50 kV



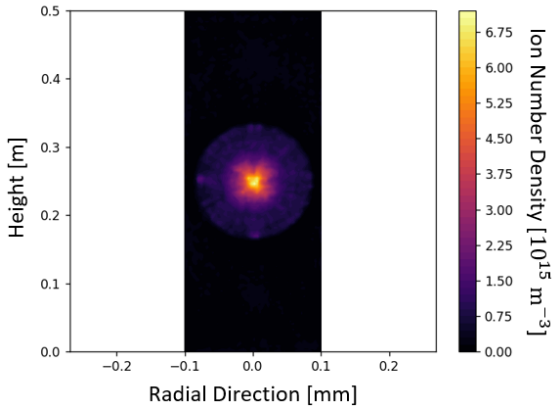


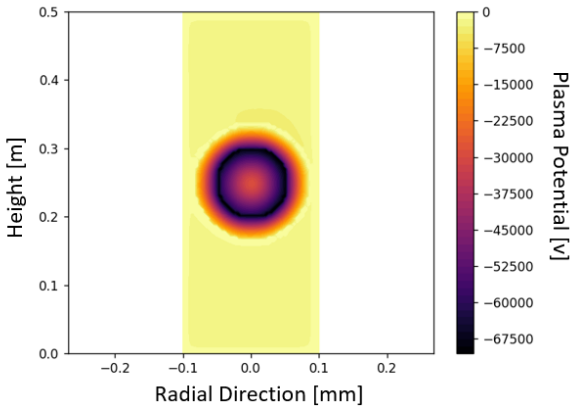
-60 kV



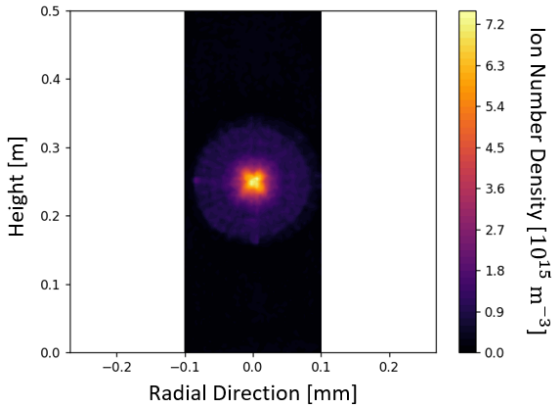


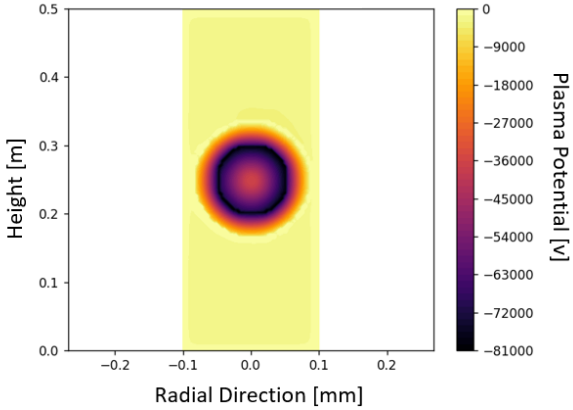
-70 kV



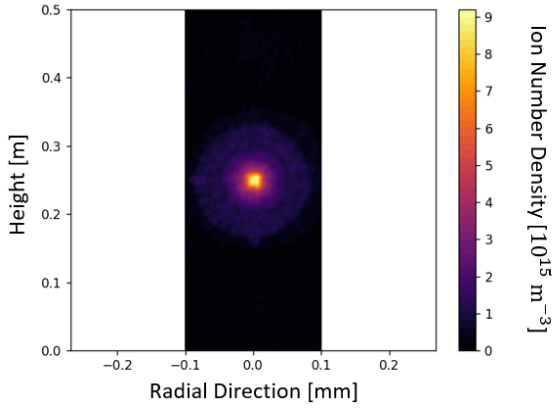


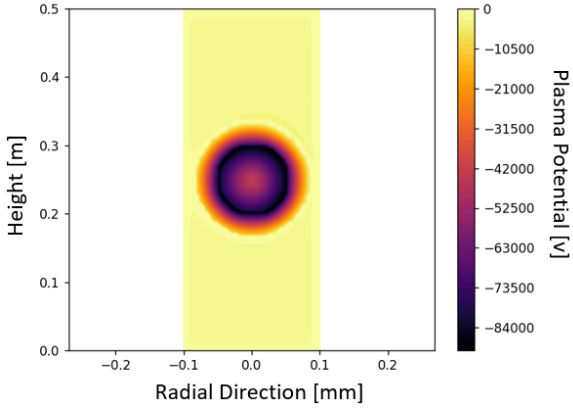
-80 kV



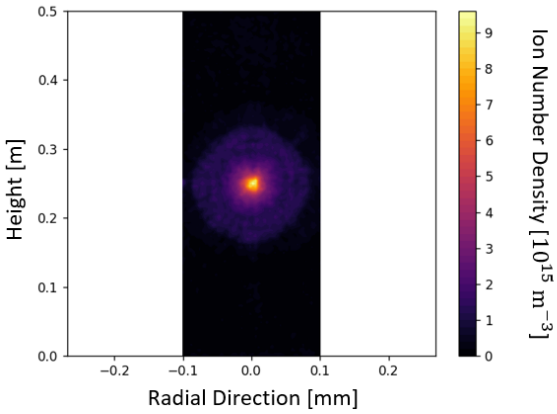


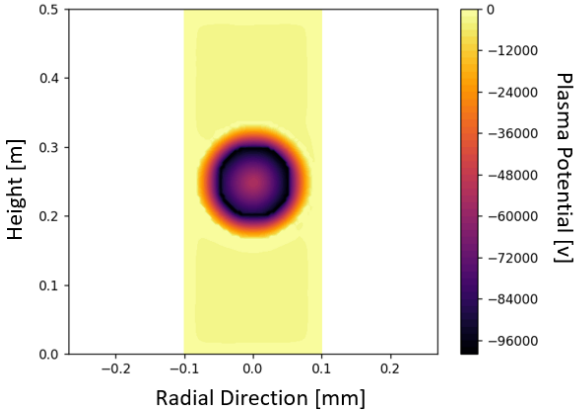
-90 kV



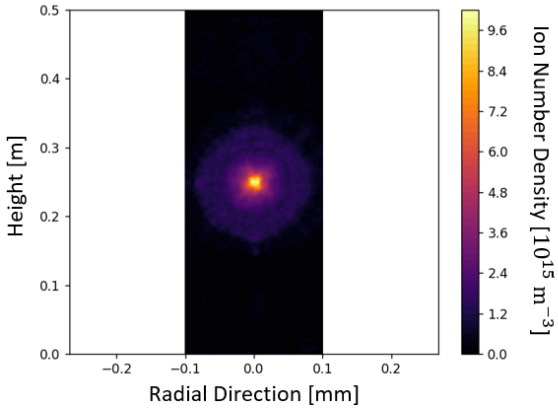


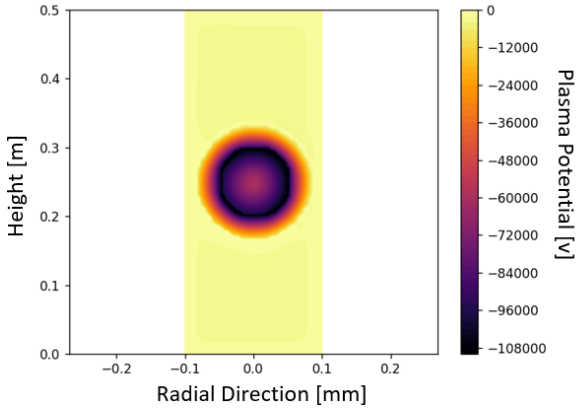
-100 kV



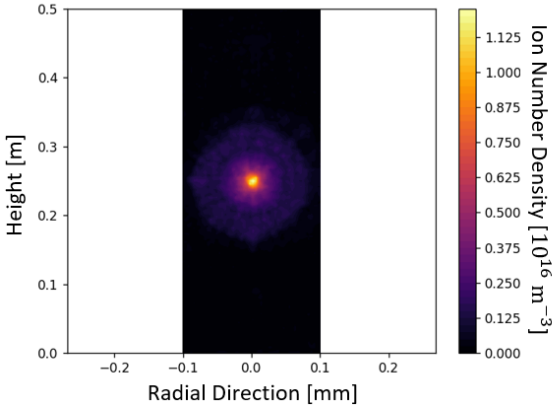


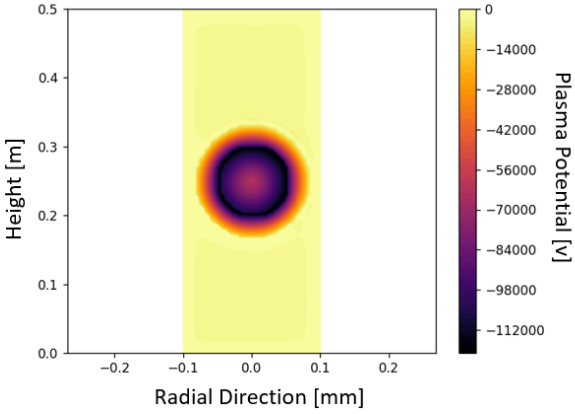
-110 kV



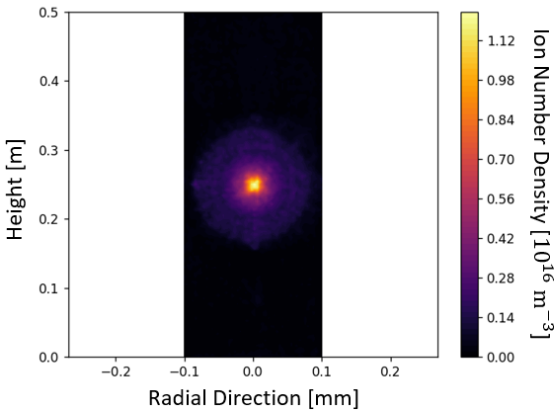


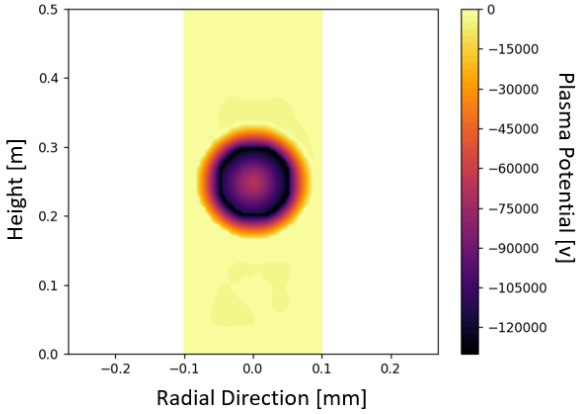
-120 kV



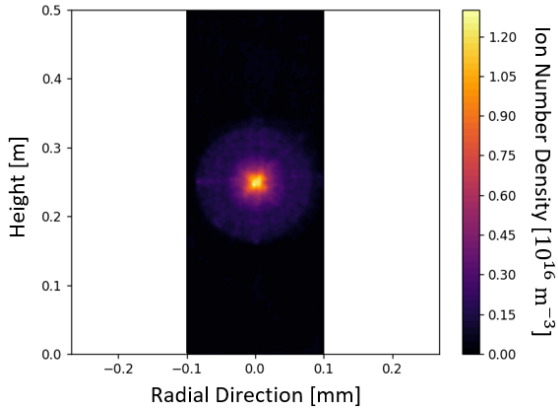


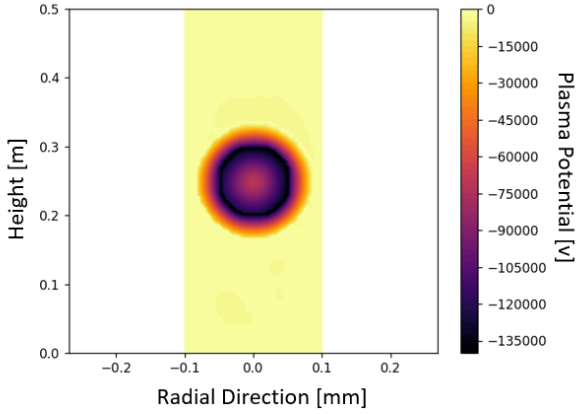
-130 kV



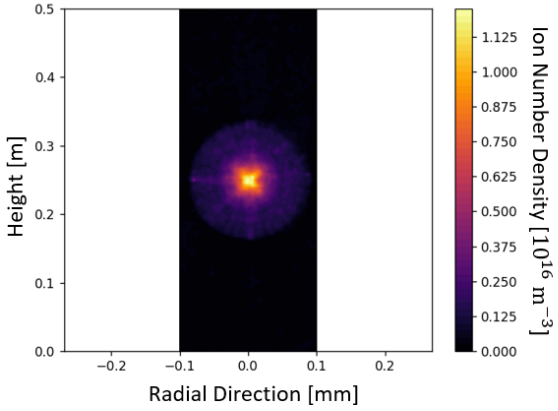


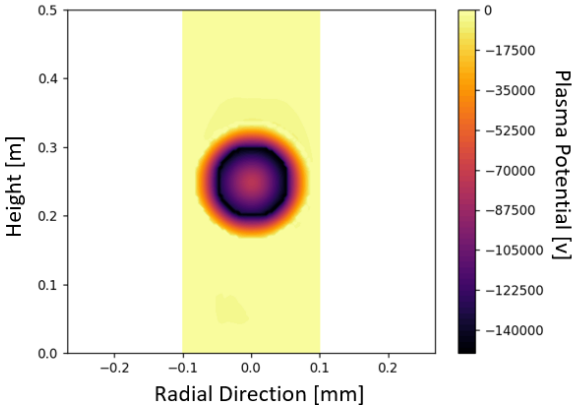
-140 kV



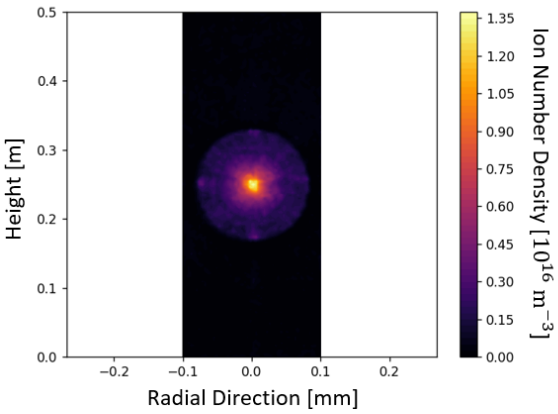


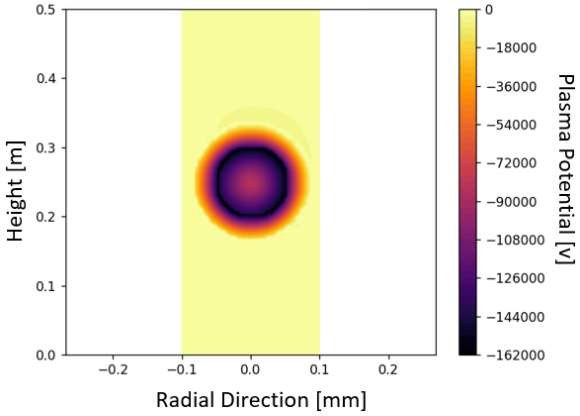
-150 kV



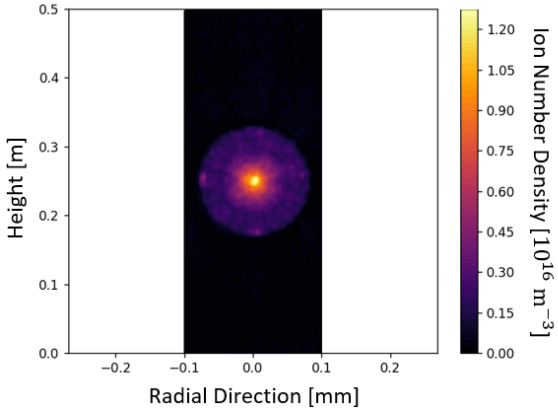


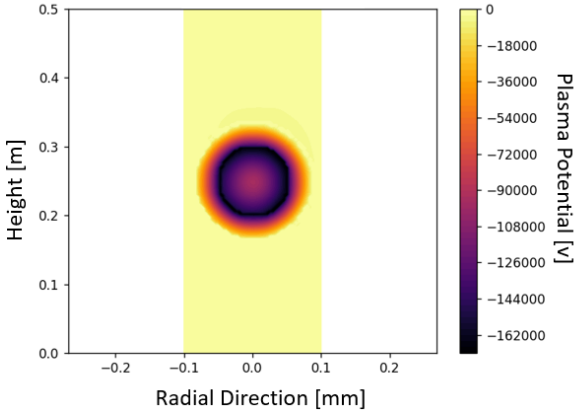
-160 kV



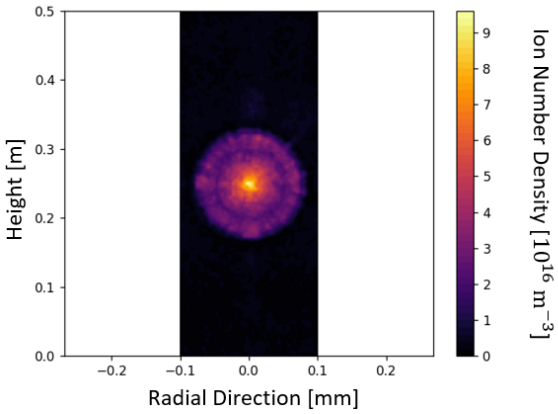


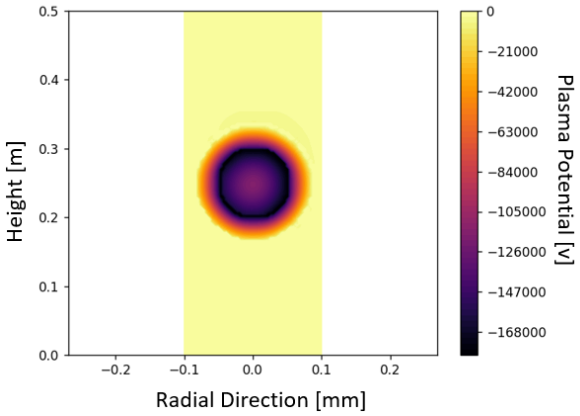
-170 kV



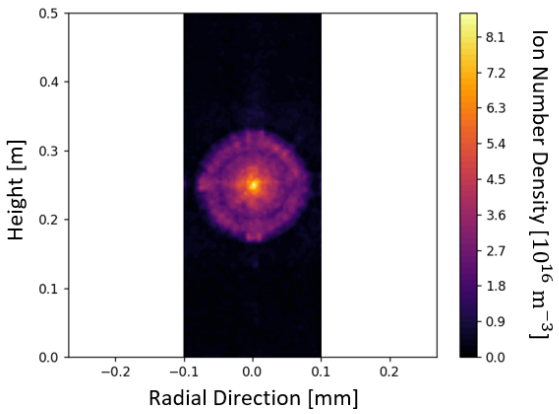


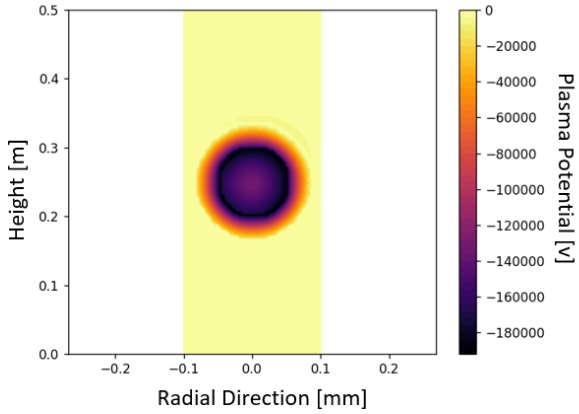
-180 kV



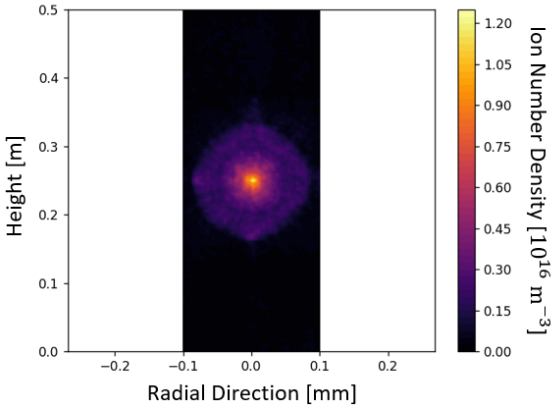


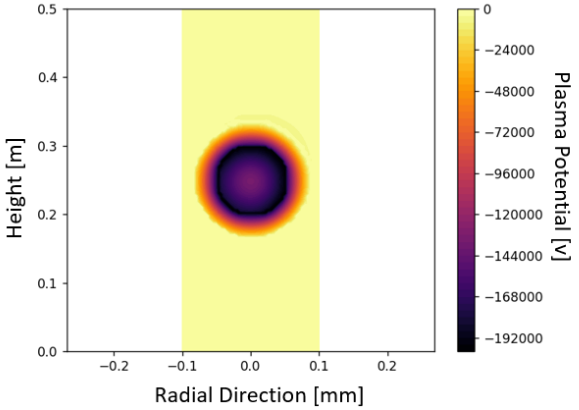
-190 kV


Height [m]
Radial Direction [mm]
Ion Number Density [10^{16} m^{-3}]


Height [m]
Radial Direction [mm]
Plasma Potential [v]

-200 kV


Height [m]
Radial Direction [mm]
Ion Number Density [10^{16} m^{-3}]


Height [m]
Radial Direction [mm]
Plasma Potential [v]

Appendix B: Particle-in-Cell Python Code

```
##########################################################################################
#     PIC Code for IEC Fusion Modeling                                                   #
#     Written by: Jacob van de Lindt                                                     #
#     Heavily modified from Lubos Brieda's solar wind on a plate MATLAB example from:  #
#     https://www.particleincell.com/2010/es-pic-method/                                 #
#                                                                                        #
#    Date: 5/10/2020                                                                     #
##########################################################################################


import numpy as np
import matplotlib.pyplot as plt
#setup constants
EPS0 = 8.854e-12       #permittivity of free space
QE = 1.602e-19         #elementary charge
kb = 1                 #boltzmann constant
AMU = 1.661e-27        #atomic mass unit
m_ion = 2*AMU          #ion mass (deuterium)

# Physical Grid Dimensions
R1 = 0.05            # Cathode [m]
GEO_C = .95          # Geometric transparency cathode (manual for now)
R2 = .085            # Anode [m]
GEO_A = .95          # Geometric transparency anode   (manual for now)
#Physical Chamber Dimensions
R_Chamber = .1       # [m]
H_Chamber = 0.5     # [m]
r_wire = .80*1e-3 / 2 # Radius of 20 guage wire in m
#Sourse Dimension and Distrobution
R_Sourse_min = .08  # smallest radius a sourse particle can appear at
sourse_spread = .0025 # largest radius a sourse particle can appear minus the minimum

#input settings
# n0 = 4.6e13
n0 = 1e16                 #electron background density in #/m^3
```

```
phi_Cathode = -210000        #cathode potential
phi0 = 0
# Te = np.abs(phi_Cathode)#reference potential
Te = 600    #electron temperature in eV
Ti = 0.1                    #ion velocity in eV (not used yet)
vth = np.sqrt(2*QE*Ti/m_ion)   #thermal velocity with Ti in eV
Operating_Pressure = 7      # Pa (Not used yet)
#calculate plasma parameters
#lD = np.sqrt(EPS0*Te/(n0*QE))      #Debye length (not used yet)
vth = np.sqrt(2*QE*Ti/m_ion)        #Thermal velocity with Ti in eV (not used yet)
lD = 0.004   # MANUAL SETTING FOR TESTING
dx = lD               #x cell size
dy = lD               #y cell size
#set simulation domain in the x dimension
nx = np.floor((2*R_Chamber)/dx).astype(int) #number of nodes in x direction, with a physical
                                            #domain +/- R_Chamber
#set simulation domain in the y dimension
ny = np.floor((H_Chamber)/dy).astype(int)   #number of nodes in y direction, with a physical
                                            #domain +/- H_Chamber / 2
ts = 800                                    #number of time steps

#calculate maximum expected velocity and timestep
E_av = (np.abs(phi_Cathode) - 0) / (R2 - R1)
a_av = E_av*QE / m_ion
v_max = np.sqrt(2*a_av*R2)
dt = 1e-10     #time step size, at vmax move 0.10dx

# Domain length in the x direction
Lx = (nx-1)*dx
# Domain length in the y direction
Ly = (ny-1)*dy
# SET UP THE POINTS REPRESENTING THE GRIDS
def get_discrete_Value(y, dy):
    # Takes in an array y and a grid spacing dy and returns an array
    # with the same physical dimensions as y but with a spacing of dy.
    actual = y / dy
    index = np.round(actual)
    return index*dy
```

```
# CATHODE SETUP
Thetav = np.arange(0, 2*np.pi, np.pi/1000)
x_cathode = R1*np.cos(Thetav)
y_cathode = R1*np.sin(Thetav)
X_cathode = get_discrete_Value(x_cathode, dx)
Y_cathode = get_discrete_Value(y_cathode, dy)

INDEX_X_Cathode = (X_cathode/dx).astype(int)
INDEX_Y_Cathode = (Y_cathode/dx).astype(int)

# ANODE SETUP
x_anode = R2*np.cos(Thetav)
y_anode = R2*np.sin(Thetav)
X_anode = get_discrete_Value(x_anode, dx)
Y_anode = get_discrete_Value(y_anode, dy)

INDEX_X_Anode = (X_anode/dx).astype(int)
INDEX_Y_Anode = (Y_anode/dy).astype(int)

# LEFT WALL SETUP
y_left_wall = np.arange(-H_Chamber/2, H_Chamber/2, dy)
x_left_wall = np.ones(y_left_wall.shape[0]) * (-R_Chamber)
INDEX_X_left_wall = (x_left_wall/dx).astype(int)
INDEX_Y_left_wall = (y_left_wall/dy).astype(int)
# RIGHT WALL SETUP
y_right_wall = np.arange(-H_Chamber/2, H_Chamber/2, dy)
x_right_wall = np.ones(y_right_wall.shape[0]) * (R_Chamber - dx)
INDEX_X_right_wall = (x_right_wall/dx).astype(int)
INDEX_Y_right_wall = (y_right_wall/dy).astype(int)
# BOTTOM WALL SETUP
x_bottom_wall = np.arange(-R_Chamber, R_Chamber, dx)
y_bottom_wall = np.ones(x_bottom_wall.shape[0]) * (-H_Chamber/2)
INDEX_X_bottom_wall = (x_bottom_wall/dx).astype(int)
INDEX_Y_bottom_wall = (y_bottom_wall/dy).astype(int)
# TOP WALL SETUP
x_top_wall = np.arange(-R_Chamber, R_Chamber, dx)
y_top_wall = np.ones(x_bottom_wall.shape[0]) * (H_Chamber/2 - dy)
```

```
INDEX_X_top_wall = (x_top_wall/dx).astype(int)
INDEX_Y_top_wall = (y_top_wall/dy).astype(int)

# Deleate repeate XY Pairs
Cathode_Id1 = np.zeros([2, INDEX_X_Cathode.shape[0]])
Cathode_Id1[0, :] = INDEX_X_Cathode
Cathode_Id1[1, :] = INDEX_Y_Cathode
Cathode_Id2 = np.unique(Cathode_Id1, axis=1)
INDEX_X_Cathode = Cathode_Id2[0, :].astype(int)
INDEX_Y_Cathode = Cathode_Id2[1, :].astype(int)

# Calculate specific weight and prepair to insert particles
np_insert = 400                      #insert 2 particles per anode cell.
#flux = 4.6e22
flux = 1.6e21*np.abs(phi_Cathode)/100000    #Flux of entering ions [ions per second]
npt = flux*dt
spwt = npt/np_insert                    #specific weight, real particles per macroparticle
mp_q = 1                                #macroparticle charge
max_part = 200000                       #buffer size
#allocate particle array
part_x = np.zeros([max_part,2]) #particle positions
part_v = np.zeros([max_part,2]) #particle velocities
sourse_storage = np.zeros([max_part,2]) #particle sourses
# Define the potential solver function

PHI_B = np.zeros([nx, ny])
idx = np.round(nx/2).astype(int)
idy = np.round(ny/2).astype(int)

#Potential Solver

def get_Potential(PHI_B, den, nx, ny, iters):
    for k in range(iters):
        PHI_OLD = PHI_B
        for i in range(1,nx-2):
            for j in range(1, ny-2):
                ni = den[i,j]
                rho = QE*(ni - n0*np.exp((PHI_OLD[i,j] - phi0)/(kb*Te))) / EPS0
```

```
                chrg = -rho*dx**2
                PHI_B[i,j] = (chrg - PHI_B[i+1, j] - PHI_B[i-1, j] - PHI_B[i, j+1] - PHI_B[i, j-
1])/(-4)
        PHI_B[INDEX_X_Cathode + idx, INDEX_Y_Cathode + idy] = phi_Cathode
        PHI_B[INDEX_X_Anode + idx, INDEX_Y_Anode + idy] = phi0
    return PHI_B


# Ion Source
def sample_Source(nump, R_S1, r_spread):
    # Updated to generate particles randomly in theta and in R.
    xv = np.zeros([nump])
    yv = np.zeros([nump])
    for i in range(nump):
        R_S = R_S1 + np.random.rand()*r_spread
        theta = np.random.rand(1)*2*np.pi   # Generate random polar angle
        x = R_S*np.cos(theta)                # Get x position
        y = R_S*np.sin(theta)                # Get y position
        xv[i] = x
        yv[i] = y
    return np.array([xv, yv])


# D-D Cross Section from 5 parameter fit
def fusion_Cross_Section(vx, vy):
    # Takes in velocity componants in m/s and returns a cross section in barns
    E = .5*m_ion*(vx**2 + vy**2)
    E = E*6.242e15  # convert J to KeV
    A1 = 46.097
    A2 = 372
    A3 = 4.36e-4
    A4 = 1.22
    A5 = 0
    AA1 = 47.88
    AA2 = 482
    AA3 = 3.08e-4
    AA4 = 1.177
    AA5 = 0
    term1 = A5 + A2/((A4 - A3*E)**2 + 1)
    term2 = E*(np.exp(A1/np.sqrt(E)) - 1)
```

```
    term3 = AA5 + AA2/((AA4 - AA3*E)**2 + 1)
    term4 = E*(np.exp(AA1/np.sqrt(E)) - 1)
    sig1 = term1/term2
    sig2 = term3/term4
    return sig1 + sig2


##############
# MAIN LOOP
############

#INITIALIZE
PHI_M = np.zeros([nx, ny])
idx = np.round(nx/2).astype(int)
idy = np.round(ny/2).astype(int)
PHI_M[INDEX_X_Cathode + idx, INDEX_Y_Cathode + idy] = phi_Cathode
PHI_M[INDEX_X_Anode + idx, INDEX_Y_Anode + idy] = phi0
fuse_pos_x = np.array([])
fuse_pos_y = np.array([])
fuse_time = np.array([])
col_counter = 0
top_counter = 0
bot_counter = 0
left_counter = 0
right_counter = 0
anode_counter = 0
cathode_counter = 0
fuse_counter = 0
num_p = 0                          #Clear number of particles
iters = 600                        #Number of iterations used in the potential solver
P_FUS = 0 # set a checker to report max fusion probability
print('Beginning Main Loop. This could take a while. \n')

for it in range(ts):
    P_FUS = 0
    print('Time Step ', it, 'Particles ', num_p)
    #reset field quantities
    den = np.zeros([nx,ny])              #number density
```

```
efx = np.zeros([nx,ny])              #electric field, x-component
efy = np.zeros([nx,ny])              #electric field, y-component
chg = np.zeros([nx,ny])              #charge distribution
col_counter = 0

# *** 1. Calculate Charge Density ***
# deposit charge to nodes
for p in range(num_p):                       #loop over particles
    fi = (part_x[p, 0] + R_Chamber-dx)/dx    #real i index of particle's cell
    i = np.floor(fi).astype(int)             #integral part
    hx = fi - i                              #the remainder
    fj = (part_x[p,1] + (H_Chamber/2)-dx)/dx #real i index of particle's cell
    j = np.floor(fj).astype(int)             #integral part
    hy = fj - j                              #the remainder
    #interpolate charge to nodes
    chg[i, j] = chg[i, j] + (1-hx)*(1-hy)
    chg[i+1, j] = chg[i+1, j] + hx*(1-hy)
    chg[i, j+1] = chg[i, j+1] + (1-hx)*hy
    chg[i+1, j+1] = chg[i+1, j+1] + hx*hy

# Calculate the Density
den = spwt*mp_q*chg / (dx**2)
den[0,:] = 2*den[0,:]              # Double density since only half volume contributing
den[nx-1,:] = 2*den[nx-1,:]
den[:,0] = 2*den[:,0]
den[:,ny-1] = 2*den[:,ny-1]
den = den + 1e3                    # Add density floor to help the solver

# *** 2. Calculate the Potential
PHI_M = get_Potential(PHI_M, den, nx, ny, iters)
print('Potential Solution Complete.')
# *** 3. Calculate the Electric Field ***
efx[1:nx-2,:] = PHI_M[0:nx-3,:] - PHI_M[2:nx-1,:]   #central difference on internal nodes
efy[:,1:ny-2] = PHI_M[:,0:ny-3] - PHI_M[:,2:ny-1]   #central difference on internal nodes
efx[0,:] = 2*(PHI_M[0,:] - PHI_M[1,:])             #forward difference on x=0
efx[nx-1,:] = 2*(PHI_M[nx-2,:] - PHI_M[nx-1,:])     #backward difference on x=Lx
efy[:,0] = 2*(PHI_M[:,0] - PHI_M[:,1])              #forward difference on y=0
efy[:,ny-1] = 2*(PHI_M[:,ny-2] - PHI_M[:,ny-1])     #forward difference on y=Ly
```

```
efx = efx / (dx**2)                                    #divide by denominator
efy = efy / (dx**2)

# *** 4. Generate New Particles
print('Generating Particles')
#insert particles randomly distributed and within the sourse region
Posv = sample_Source(np_insert, R_Sourse_min, sourse_spread)
part_x[num_p:num_p+np_insert, 0] = Posv[0,:]   #x position
part_x[num_p:num_p+np_insert, 1] = Posv[1,:]   #y position
#sourse_storage[num_p:num_p+np_insert, 0] = Posv[0,:]
#sourse_storage[num_p:num_p+np_insert, 1] = Posv[1,:]
#sample maxwellian in x and y
pt1 = np.random.rand(np_insert)
pt2 = np.random.rand(np_insert)
pt3 = np.random.rand(np_insert)
pt11 = np.random.rand(np_insert)
pt12 = np.random.rand(np_insert)
pt13 = np.random.rand(np_insert)
part_v[num_p:num_p+np_insert,0] = (-1.5 + pt1 + pt2 + pt3)*vth     # x velocity
part_v[num_p:num_p+np_insert,1] = (-1.5 + pt11 + pt12 + pt13)*vth  # y velcoity
num_p = num_p + np_insert                                          #increment particle counter
# *** Move Particles ***
print('Moving Particles...   DO NOT INTERRUPT')
p=0
while p < num_p:        # Loop over particles
    fi = (part_x[p, 0] + R_Chamber-dx)/dx   # i index of particle's cell. Taking into account
    i = np.floor(fi).astype(int)         # that the physical domain is centered at (0,0)
    hx = fi - i
    fj = (part_x[p,1] + (H_Chamber/2)-dx)/dx
    j = np.floor(fj).astype(int)
    hy = fj-j
    #print('Fi: ', fi, 'Fj: ', fj)
    # gather electric field
    E = np.array([0,0])
    E = np.array([efx[i,j], efy[i,j]])*(1-hx)*(1-hy)      #contribution from (i,j)
    E = E + np.array([efx[i+1,j], efy[i+1,j]])*hx*(1-hy)            #(i+1,j)
    E = E + np.array([efx[i,j+1], efy[i,j+1]])*(1-hx)*hy            #(i,j+1)
    E = E + np.array([efx[i+1,j+1], efy[i+1,j+1]])*hx*hy            #(i+1,j+1)
```

```
#print('E:', E)
# Update Velocity and Position
F = QE*E      # Lorenz force F = qE
a = F/m_ion   # Acceleration
part_v[p,:] = part_v[p,:] + a*dt
part_x[p,:] = part_x[p,:] + part_v[p,:]*dt
#print(part_v[p,:])
#print(part_x[p,:])
# Get Fusion probability
vx = part_v[p, 0]
vy = part_v[p, 1]
delx = vx * dt * 1e4
dely = vy * dt * 1e4
path_len = np.sqrt(delx**2 + dely**2)
sigma = fusion_Cross_Section(vx, vy)*1e-28 # microscopic cross section [m^2]

# gather density at particle position
Rho = den[i,j]*(1-hx)*(1-hy)      #contribution from (i,j)
Rho = Rho + den[i+1,j]*hx*(1-hy)            #(i+1,j)
Rho = Rho + den[i,j+1]*(1-hx)*hy            #(i,j+1)
Rho = Rho + den[i+1,j+1]*hx*hy            #(i+1,j+1)

# Calculate macroscopic cross section
SIGMA = Rho * sigma
Prob_fusion = SIGMA * path_len * spwt
# Store
if Prob_fusion > P_FUS:
    P_FUS = Prob_fusion
# Prepare to check for fusion
random = np.random.rand(1)

# Process Boundries and Sinks
R = np.sqrt(part_x[p,0]**2 + part_x[p,1]**2)
#print('Finished Position and Velocity Update')
#print('Checking Top Wall')
# Top Wall
if part_x[p,1] > H_Chamber/2:
    part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
```

```
    part_v[p,:] = part_v[num_p-1,:]
    part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
    part_v[num_p-1,:] = np.array([0,0])
    num_p = num_p - 1              # Reduce particle count
    p = p - 1                      # Reduce particle index
    col_counter = col_counter + 1
    top_counter = top_counter + 1

# Bottom Wall
elif part_x[p,1] < -H_Chamber/2:
    #print('Checking Bottom Wall')
    part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
    part_v[p,:] = part_v[num_p-1,:]
    part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
    part_v[num_p-1,:] = np.array([0,0])
    num_p = num_p - 1              # Reduce particle count
    p = p - 1                      # Reduce particle index
    col_counter = col_counter + 1
    bot_counter = bot_counter + 1

# Right Wall
elif part_x[p,0] > R_Chamber:
    #print('Checking Right Wall')
    part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
    part_v[p,:] = part_v[num_p-1,:]
    part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
    part_v[num_p-1,:] = np.array([0,0])
    num_p = num_p - 1              # Reduce particle count
    p = p - 1                      # Reduce particle index
    col_counter = col_counter + 1
    right_counter = right_counter + 1

# Left Wall
elif part_x[p,0] < -R_Chamber:
    #print('Checking Left Wall')
    part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
    part_v[p,:] = part_v[num_p-1,:]
    part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
```

```
    part_v[num_p-1,:] = np.array([0,0])
    num_p = num_p - 1              # Reduce particle count
    p = p - 1                      # Reduce particle index
    col_counter = col_counter + 1
    left_counter = left_counter + 1

# Process grids

# Anode
elif (R < R2 + r_wire) and (R > R2 - r_wire):
    #print('Check Anode')
    prob = np.random.rand(1)
    if prob > GEO_A:
        # Delete particle if it collides with the grid
        part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
        part_v[p,:] = part_v[num_p-1,:]
        part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
        part_v[num_p-1,:] = np.array([0,0])
        num_p = num_p - 1              # Reduce particle count
        p = p - 1                      # Reduce particle index
        col_counter = col_counter + 1
        anode_counter = anode_counter + 1

# Cathode
elif (R < R1 + r_wire) and (R > R1 - r_wire):
    #print('Check Cathode')
    prob = np.random.rand(1)
    if prob > GEO_A:
        # Delete particle if it collides with the grid
        part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
        part_v[p,:] = part_v[num_p-1,:]
        part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
        part_v[num_p-1,:] = np.array([0,0])
        num_p = num_p - 1              # Reduce particle count
        p = p - 1                      # Reduce particle index
        col_counter = col_counter + 1
        cathode_counter = cathode_counter + 1
```

```
        elif random <= Prob_fusion:
            print('FUSION!\n')
            fuse_pos_x = np.append(fuse_pos_x, part_x[p, 0])
            fuse_pos_y = np.append(fuse_pos_y, part_x[p, 1])
            fuse_time = np.append(fuse_time, dt*it)
            fuse_counter = fuse_counter + 1
            # Delete particle if it fused
            part_x[p,:] = part_x[num_p-1,:]  # Kill particle by replacing with last particle
            part_v[p,:] = part_v[num_p-1,:]
            part_x[num_p-1,:] = np.array([0,0]) # Reset last particle to 0
            part_v[num_p-1,:] = np.array([0,0])
            num_p = num_p - 1              # Reduce particle count
            p = p - 1                      # Reduce particle index

        p = p + 1                              # Move to the next particle
    print('Finished Moving Particles.')
    print('Net Change in Ion Population: ', np_insert - col_counter)
    print(col_counter, ' particles lost.')
    print('Max probability of fusion: ', P_FUS, '\n')

# Replace voltage with Phi_Cathode
np.savetxt('Density_210kV_1.csv', den, delimiter=',')
np.savetxt('Potential_210kV_1.csv', PHI_M, delimiter=',')
np.savetxt('fus_time_210kV_1.csv', fuse_time, delimiter=',')
np.savetxt('fus_pos_x_210kv_1.csv', fuse_pos_x, delimiter=',')
np.savetxt('fus_pos_y_210kV_1.csv', fuse_pos_y, delimiter=',')
np.savetxt('part_x_210kV_1.csv', part_x, delimiter=',')
np.savetxt('part_v_210kV_1.csv', part_v, delimiter=',')

av_fusion_rate = (fuse_counter / (dt*ts)) / 1e4
print('Field Properties and Fusion Data Saved to File.\n')
print('Average Fusion Rate: %.6f Fusions per Second\n' % av_fusion_rate)
```