PIC Modeling of Coulomb Collisional Effects in a Wall-Bounded Plasma

Jordan DeHaven

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

University of Washington

2020

Committee:
Uri Shumlak
Eric Meier

Program Authorized to Offer Degree:
Aeronautics and Astronautics

University of Washington

Abstract

PIC Modeling of Coulomb Collisional Effects in a Wall-Bounded Plasma

Jordan DeHaven

Chair of the Supervisory Committee:
Uri Shumlak
Department of Aeronautics and Astronautics

Collisions between particles in a hot plasma can be ignored in many high-temperature plasma simulations. However, at lower temperatures, collisional effects play a significant role in determining the plasma's overall behavior. While fluid modeling can accurately capture collisional effects (with some limitations), proper definition of boundary conditions for fluid modeling depend on accurate assumptions about the kinetic (real) behavior of plasma near material surfaces, where the plasma sheath dominates the plasma behavior. This thesis project is an attempt to develop a tool to elucidate sheath and presheath plasma behavior in the presence of strong collisional effects. Using the Particle In Cell (PIC) method, a Coulomb collisional model is implemented to study near-surface plasma behavior. The information gleaned from this project could provide important refinement of boundary conditions for fluid codes, and more generally, insights into kinetic behavior of the collisional plasma sheath.

Contents

1 Introduction

Computational modeling is best used as a complement to experiment and theory. A simulation can be used to verify a theoretical prediction or inform a future experiment. This thesis project is an attempt at both.

A plasma may be represented computationally as either a fluid or a collection of individual particles. These alternate schemes are known as fluid and kinetic models, respectively. In a fluid model, collective plasma behavior is represented, rather than the behavior of individual particles, as in a kinetic model. While this can significantly reduce computation time in many cases, the fluid equations rely on the plasma velocity distribution to be near-Maxwellian. In some regions, the distribution may not be Maxwellian. One such example is near the plasma-wall interface. Fusion plasmas exist at tem-

peratures upwards of 100 million Kelvin. As one could imagine, great effort is expended in devising ways to keep the hot plasma away from the walls of its containment vessel. The region that develops between the wall and the bulk plasma is called the sheath. As the number of plasma particles drops from the edge of the bulk plasma to the wall, the assumption that the velocity distribution of those particles is Maxwellian breaks down. This is why many fluid models have domains that span the bulk plasma alone and ignore the sheath.

Useful information can be gleaned from studying the plasma sheath. For example, computational models that utilize the fluid aspect of a plasma require the flux of energy through the sheath-edge as a boundary condition. Section 2 explores the energy flux at the sheath edge from a theoretical stand point. Standard, fluid, and kinetic methods for calculating this value are discussed.

At high plasma temperatures, collisional effects between constituent particles is insignificant enough to be ignored in computational models. However, when temperatures are low, collisional effects are critical to modeling a plasma accurately.

A kinetic model of a plasma provides an accurate representation of a plasma because it does not rely on assumptions about the distribution of the properties of its particles. A kinetic model will capture whatever distribution occurs, providing a more accurate representation of reality. In Section 3, I describe the Particle-in-Cell (PIC) kinetic model used to investigate the sheath edge behavior at low temperatures. Finally, in Section 4, the energy flux results are discussed.

2 Physics at the Boundary

2.1 Standard Fluid Energy Flux Density

The electron energy flux density at the sheath edge may be derived by considering the one-way heat flux density for a Maxwellian and the potential energy across the sheath. All energy flux quantities will pertain to electrons, so any 'e' sub- or superscript will be omitted to reduce textual clutter.

To find the one-way heat flux density for a Maxwellian, following Stangeby[1], we begin with the probability distribution function for a Maxwellian velocity distribution,

$$f(\mathbf{v}) = n\left(\frac{\beta}{\pi}\right)^{3/2} e^{-\beta \mathbf{v}^2}, \tag{2.1}$$

where n is the species density at a given point in physical space, $\beta = m/2kT$, and $\mathbf{v}^2 = v_x^2 + v_y^2 + v_z^2$.

Electrons with 1D velocity passing through a sheath edge have particle flux density,

$$\Gamma_x = \int_{v_x}\int_{v_y}\int_{v_z} v_x f(\mathbf{v}) dv_x dv_y dv_z. \tag{2.2}$$

All velocities are considered, so (2.2) as a definite integral becomes

$$\Gamma_x = \int_{v_x=0}^{\infty}\int_{v_y=-\infty}^{\infty}\int_{v_z=-\infty}^{\infty} v_x f(\mathbf{v}) dv_x dv_y dv_z. \tag{2.3}$$

Plugging (2.1) into (2.3), we have the expression,

$$\Gamma_x = n\left(\frac{\beta}{\pi}\right)^{3/2} \int_{v_x=0}^{\infty}\int_{v_y=-\infty}^{\infty}\int_{v_z=-\infty}^{\infty} v_x e^{-\beta(v_x^2+v_y^2+v_z^2)} dv_x dv_y dv_z, \tag{2.4}$$

which becomes

$$\Gamma_x = \frac{1}{4} n\bar{c}, \tag{2.5}$$

where $\bar{c} = (8kT/\pi m)^{1/2}$ is the average particle velocity in the sheath. It should be noted that this one-way particle flux density is for an ordinary, non-drifting Maxwellian, which is a good approximation for the electrons reaching the wall, since their mean velocity is much larger than their bulk velocity.

Now, we can arrive at the one-way heat flux density:

$$q = E_{kin}\Gamma_x \tag{2.6}$$

where $E_{kin} = \frac{1}{2}m(v_x^2 + v_y^2 + v_z^2)$ is the kinetic energy of each particle. Thus,

$$q = \int_{v_x=0}^{\infty}\int_{v_y=-\infty}^{\infty}\int_{v_z=-\infty}^{\infty} \frac{1}{2} m(v_x^2 + v_y^2 + v_z^2) v_x f(\mathbf{v}) dv_x dv_y dv_z. \tag{2.7}$$

$$q = 2kT\Gamma_x \tag{2.8}$$

In general, (2.8) can be used to determine the forward going flux for a Maxwellian at any location between the bounds of a domain. Since our interest lies in the particles that have a high enough energy to reach a wall, (2.8) can be applied to the flux of electron energy to the wall,

$$q_{sw} = 2T\Gamma_{se}, \tag{2.9}$$

where the sw and se subscripts refer to "sheath-wall interface" and "sheath edge", respectively. Plasma temperature will be expressed in units of eV, such that Boltzmann's constant, k, is absorbed into T. Temperature is calculated by taking moments of the distribution function

$$T = \frac{1}{n}\int \frac{m_e}{3}\mathbf{v}^2 f(\mathbf{v})d\mathbf{v} = \frac{m_e}{3}\left\langle \mathbf{v}^2 \right\rangle. \tag{2.10}$$

The quantity derived in (2.9) is independent of the fact that electrons which reach the wall are adsorbed, thereby leaving a net positive charge near the wall. This adsorption causes a potential drop, ϕ, between the plasma and the charged wall. Given without derivation, the heat flux density associated with this potential drop is

$$q_\phi = |e\phi|\Gamma_{se} \approx 3T\Gamma_{se}. \tag{2.11}$$

Finally, by summing (2.9) and (2.11), we arrive at the total energy flux density at the sheath edge,

$$h_{se} = q_{sw} + q_\phi \approx (2T + 3T)\Gamma_{se} = 5T\Gamma_{se}. \tag{2.12}$$

In the results discussed in Section 4, the quantity calculated in (2.12) will be relabeled to h_{std}, in order to distinguish this standard fluid energy flux density from that which is calculated via kinetic methods.

2.2 Kinetic Definition of Energy Flux Density

Braginksii (1965)[2] derived an expression for the total energy flux density for the electron fluid. Here, we apply it to the sheath edge. This definition is written as:

$$h_k = \left(\frac{nm}{2}V^2 + \frac{5}{2}nT\right)V_x + q_k, \tag{2.13}$$

where V, n, and T are the mean velocity, density, and temperature, respectively. The 'k' subscript refers to that which is calculated using kinetic methods. The first and second terms represent the kinetic and internal energy fluxes, respectively. The third term, the heat flux density, q_k, is the one dimensional energy carried across the sheath edge that is associated with the random motion of the particles. It is defined as

$$q_k = \int \frac{m}{2}v'^2\mathbf{v}f(t,\mathbf{x},\mathbf{v})d\mathbf{v} = nm\left\langle\frac{v'^2}{2}\mathbf{v}\right\rangle, \tag{2.14}$$

where $\mathbf{v}' = \mathbf{v} - \mathbf{V}$. For our one dimensional case, (2.13) can be rewritten:

$$q_k = nm\left\langle\frac{v'^2}{2}v_x\right\rangle \tag{2.15}$$

The heat flux q can also be approximated, for fluid modeling purposes, in terms of the thermal conductivity and the temperature gradient across a region of the domain,

$$q_f = -\kappa_\parallel \nabla T. \tag{2.16}$$

where the 'f' subscript refers to the fluid method for calculating this heat flux. Expressed in SI units[1],

$$\kappa_\parallel = \frac{30692}{Z_i \ln\Lambda}T^{5/2} \tag{2.17}$$

The parallel thermal conductivity is used because we consider an unmagnetized plasma. Since our interests lie in calculating the heat flux at the sheath edge, we take the temperature gradient between cells across the sheath edge. The heat flux in (2.16) can be used in place of q_k in (2.13) to calculate an h_f.

3 The Kinetic Plasma Model

3.1 Normalization Scheme

All variables are normalized in order to reduce program complexity and to represent important quantities in appropriate scales. Each of the variables,

x, t, q, m, and ϵ_0, are multiplied by a constant that cancels the units. This constant contains the characteristic value of the variable, denoted by a *naught* subscript.

$$\frac{\partial x}{\partial t} = v_x \quad \Longrightarrow \quad \frac{\partial \tilde{x}}{\partial \tilde{t}} = \tilde{v}_x \tag{3.1}$$

$$\frac{\partial v_x}{\partial t} = \frac{q}{m} E_x \quad \Longrightarrow \quad \frac{\partial \tilde{v}_x}{\partial \tilde{t}} = -\tilde{E}_x \tag{3.2}$$

$$E_x = -\frac{\partial \phi}{\partial x} \quad \Longrightarrow \quad \tilde{E}_x = -\frac{\partial \tilde{\phi}}{\partial \tilde{x}} \tag{3.3}$$

$$\frac{\partial^2 \phi}{\partial x^2} = -\frac{\rho_c}{\epsilon_0} \quad \Longrightarrow \quad \frac{\partial^2 \tilde{\phi}}{\partial \tilde{x}^2} = -\tilde{\rho}_c \tag{3.4}$$

(3.1) is the simple equation for calculating a velocity from a distance travelled over time. To normalize the velocity, the length and time variables must first be normalized. The length is normalized as follows: $\tilde{x} = Cx$, where the constant C is equal to $1/L_0$. In the same way, time is normalized: $\tilde{t} = Ct$, where $C = 1/t_0$. Velocity is therefore the normalized quotient of these two scaled variables.

(3.2) describes the acceleration of a particle due to the presence of an electric field. The velocity and time variables are normalized as before. Particle charge is normalized as follows: $\tilde{q} = Cq$, where $C = 1/q_0$. In this program, we set the characteristic charge equal to the elementary charge, $q_0 = e$, and q equal to the charge of the particle. Thus the normalized charge for an electron reduces to $\tilde{q} = -1$. Particle mass is similarly treated. Both the electron and ion masses are normalized by the characteristic electron mass. The normalized electron and ion masses reduce to $\tilde{m_e} = m_e/m_{e0} = 1$ and $\tilde{m_i} = m_i/m_{e0} \approx 1836$, respectively.

(3.3) is Gauss' Law and contains the electric potential variable ϕ. The potential is normalized as a result of E_x and x being normalized.

(3.4) is Poisson's equation and contains the vacuum permittivity, ϵ_0, which may be normalized. This normalization is not described here in order to avoid confusion between the *naught* inherent in the quantity and the *naught* that identifies its characteristic value. Since ϵ_0 is a constant throughout the simulation, just as particle mass and charge are, its normalization

reduces to 1. Charge density ρ_c is normalized as a result of the normalization of ϕ and x, but may also be independently normalized using the particle density, n:

$$\rho_c = qn \quad \Longrightarrow \quad \rho_c = -\tilde{n} \tag{3.5}$$

Particle density is normalized somewhat differently than the other variables. In general, $\tilde{n}$, is given by

$$\tilde{n} = \frac{\Sigma_{i=1}^{N} W_{N,i}}{\tilde{L}}. \tag{3.6}$$

Here, we introduce a weighting number for the purpose of counting the particles, where $W_n = \tilde{L}/N$. $\tilde{L}$ is the normalized total length of the domain and N is the total number of particles in the simulation. This results in $\tilde{n} = n/n_0 = 1$.

Based on the normalizations given, equations (3.1-3.5) can be written in terms of the characteristic quantities:

$$\frac{L_0}{t_0} = v_0 \tag{3.7}$$

$$\frac{v_0}{t_0} = \frac{e}{m_0} E_0 \tag{3.8}$$

$$E_0 = -\frac{\phi_0}{L_0} \tag{3.9}$$

$$\frac{\phi_0}{L_0^2} = \frac{\rho_0}{\epsilon_0} \tag{3.10}$$

$$\rho_0 = q_0 n_0 \tag{3.11}$$

Equations (3.6-3.10) can be combined to give the characteristic time in terms of the characteristic plasma frequency:

$$\frac{2\pi}{t_0} = \left(\frac{e^2 n_0}{m_0 \epsilon_0} \right)^{1/2} \equiv \omega_{p0} \tag{3.12}$$

Similarly, the characteristic length can be written in terms of the characteristic Debye length:

$$L_0 = v_0/\omega_{p0} = \left(\frac{k T_0 \epsilon_0}{n_0 e^2} \right)^{1/2} \equiv \lambda_{D0}, \tag{3.13}$$

where $v_0 = (kT_0/m_0)^{1/2}$.

It can be shown that the characteristic plasma frequency and characteristic Debye length are equal to their respective actual quantities:

$$\tilde{\omega}_p = \frac{\omega_p}{\omega_{p0}} = \left(\frac{e^2 n}{m\epsilon_0}\right)^{1/2} \left(\frac{m_0\epsilon_0}{e^2 n_0}\right)^{1/2} = \left(\frac{n}{n_0}\right)^{1/2} = \tilde{n}^{1/2} = 1 \tag{3.14}$$

$$\tilde{\lambda}_D = \frac{\lambda_D}{\lambda_{D0}} = \left(\frac{kT\epsilon_0}{ne^2}\right)^{1/2} \left(\frac{n_0 e^2}{kT_0\epsilon_0}\right)^{1/2} = \left(\frac{n_0 T}{nT_0}\right)^{1/2} = \left(\frac{\tilde{T}}{\tilde{n}}\right)^{1/2} = 1 \tag{3.15}$$

where $\tilde{T} = T/T_0 = 1$. Equations (3.12-2.15) show that, under our normalization scheme, time can be considered to have units of ω_p^{-1} and that length has units of λ_D.

3.2 Particle-In-Cell (PIC)

The Particle-In-Cell (PIC) method[2] is a powerful tool for modeling and studying a plasma. With it, the kinetic properties of a plasma may be fully described. PIC works by incorporating elements of electromagnetism, computational discretization, and time integration. It is composed of the following operations.

3.2.1 Grid and Particle Generation

Everyday experiences occur on a continuous time-space scale. Temperature, for example, can be thought of as a field that seamlessly permeates a volume. We do not typically consider temperature to exist at discrete points. In a computational simulation, quantities, such as temperature, can only exist at discrete points. To approximate the continuous nature of reality, we increase the density of these points, with the limiting factors being computation time and computer memory.

In this program, a one-dimensional grid is created to discretize naturally continuous quantities, such as electric field, charge density, and electric potential. Particles are given random velocities according to a Maxwellian distribution, as well as an appropriate mass and charge.

A typical presupposition of a plasma is that it is quasi-neutral, meaning that it's electron density is approximately equal to its ion density ($n_e \approx n_i$).

A background distribution of ions were created to provide this effect. For a hydrogen plasma, the charge of an ion is equal and opposite to that of an electron, but it's mass is about 1836 times more. The characteristic (thermal) speed of electrons is faster than ions by $\sqrt{1836}$, and associated timescales are much shorter. In the classical PIC method, ion motion is also modeled. To focus on the short electron timescales, and for general simplicity and computational expediency, ions are assumed to be static from the point of view of the classical PIC algorithm.

3.2.2 Equations of Motion

The equation of motion for the electrons is given by:

$$x_{new} = x_{old} + v\Delta t \tag{3.16}$$

where v is the velocity of an electron at the time of x_{old} and Δt is the time interval between x_{old} and x_{new}. The time step (Δt) is chosen to be small enough such that important physical phenomena are represented and large enough to be computationally efficient by resolving an appropriate level of detail.

3.2.3 Field Equations

The field equations describe how the electrons change their velocity and are derived from known quantities of the electron and ion distributions. First, the charge density is calculated on the grid points, according to a weighting scheme (Section ?). Then, the relevant Maxwell's equations are solved at each instant in time.

$$\nabla \cdot E = \frac{\rho}{\epsilon_0}, \tag{3.17}$$

$$E = -\nabla\phi, \tag{3.18}$$

where E is the electric field, ρ is the charge density, and ϕ is the electric potential. Equations (3.17) and (3.18) combine to give Poisson's equation,

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}. \tag{3.19}$$

Using finite differences, (3.19) and (3.18) can be rewritten as follows,

$$\begin{aligned} \frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{(\Delta x)^2} &= -\frac{\rho_j}{\epsilon_0}, \\ \mathbf{A}\phi &= -\frac{(\Delta x)^2}{\epsilon_0}\rho_j, \end{aligned} \tag{3.20}$$

$$E_j = \frac{\phi_{j-1} - \phi_{j+1}}{2\Delta x}, \tag{3.21}$$

where the subscript j denotes a particular grid point. (3.20) is solved using a finite difference coefficient matrix A. Once the electric field is procured, the force on each particle from the grid points can be calculated, using the Lorentz force law:

$$F = qE. \tag{3.22}$$

Then, using Newton's second law, the particles are accelerated,

$$\begin{aligned} a = \frac{\Delta v}{\Delta t} &= \frac{F}{m}, \\ \Delta v &= \frac{F}{m}\Delta t. \end{aligned} \tag{3.23}$$

Finally,

$$v_{new} = v_{old} + \Delta v. \tag{3.24}$$

3.2.4 Field and Particle Weighting

The charge density, ρ, is calculated on the grid points from the continuous particle distribution. A first-order weighting scheme was introduced to take into account a particle's proximity to each grid point. Figure 1 depicts a particle's weighted influence on the grid.

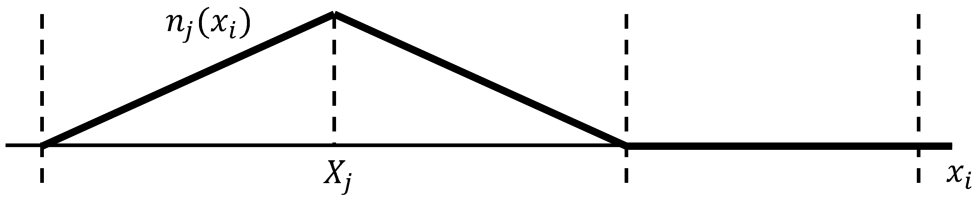


Figure 1: First-order, linear weighting for a particle, x_i, passing through grid point, X_j. Only adjacent cells contribute to the weighting.

3.2.5 Simple Harmonic Motion Test

To test the accuracy of the base PIC program, a 2-particle simple harmonic motion test was devised. Two particles were initialized along a periodic domain of length 2π. The two particles had initial positions $x = [-\pi/4, \pi/4]$ and initial velocities $v = [0, 0]$. At $t = 0$ the particles repelled each other. Since the boundary was periodic, they were repelled again as they approached the edges of the domain. This periodic motion is demonstrated in Figure 2.

The equation for position in SHM is:

$$x(t) = A\cos(\omega t) \tag{3.25}$$

where A is the position amplitude and ω is the oscillation frequency in radians per time unit. Here, one time unit is one Δt.

Equation (3.25) is solved analytically by using the amplitude and average period from the simulated results. The resulting vector is the set of positions the simulated results should have, if they indeed followed simple harmonic motion. Figure 1 depicts the close adherence of the simulated motion with SHM theory. The percent error for the motion of each particle after many time steps is 0.74%.

3.3 Floating, Conducting Walls

The simulated boundary of the domain is a pair of conducting walls. The potential at the walls is fixed at 0. Electrons that leave the bounds of the domain (adsorbed by the walls) are effectively removed from the simulation, leaving a net positive charge in both sheath regions from the excess ions. The sheath edge is defined as the cell in which the charge imbalance between

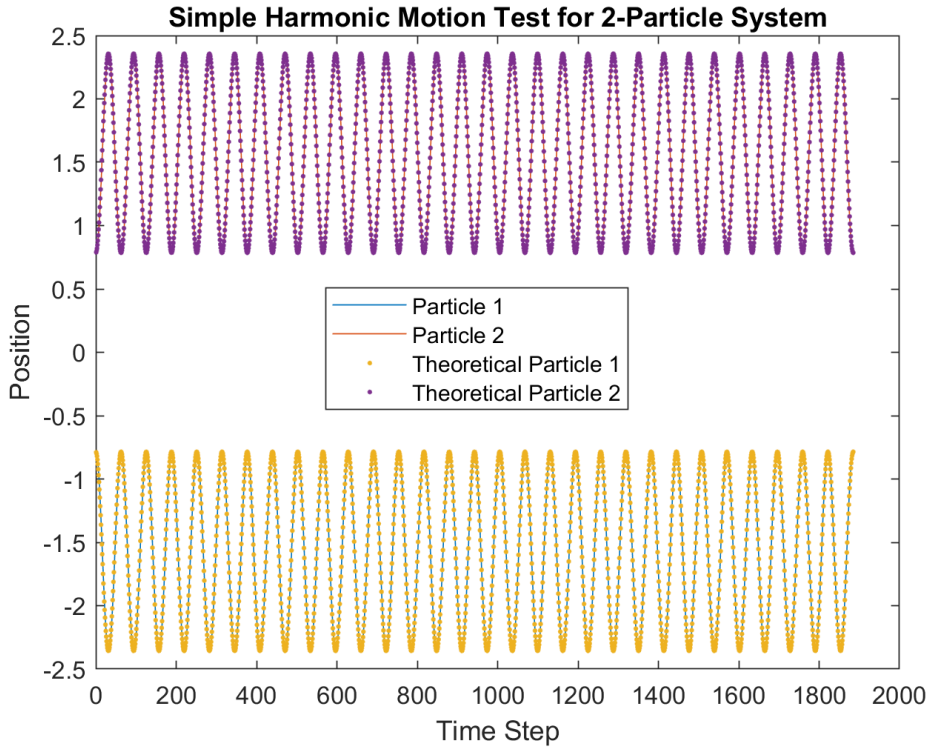


Figure 2: Simple harmonic motion exhibited by a 1-D 2-particle system. Solid curves are the simulated results. Dotted curves represent the theoretical values calculated from (3.25).

ions and electrons is 1%. In most simulations, this is about 5 Debye lengths from each wall.

Figure 3 shows the phase space (x, v_x) positions of particles, along with plots of charge density, electric potential, and electric field as functions of spatial position after 80 plasma periods of evolution. Figure 4 contains phase space plots of wall and bulk plasma regions two Debye lengths wide. The left-wall and right-wall images of Figure 4 depict the repulsion that occurs when a net positive charge is built up near a wall. Once steady state is reached, no more particles leave the domain.

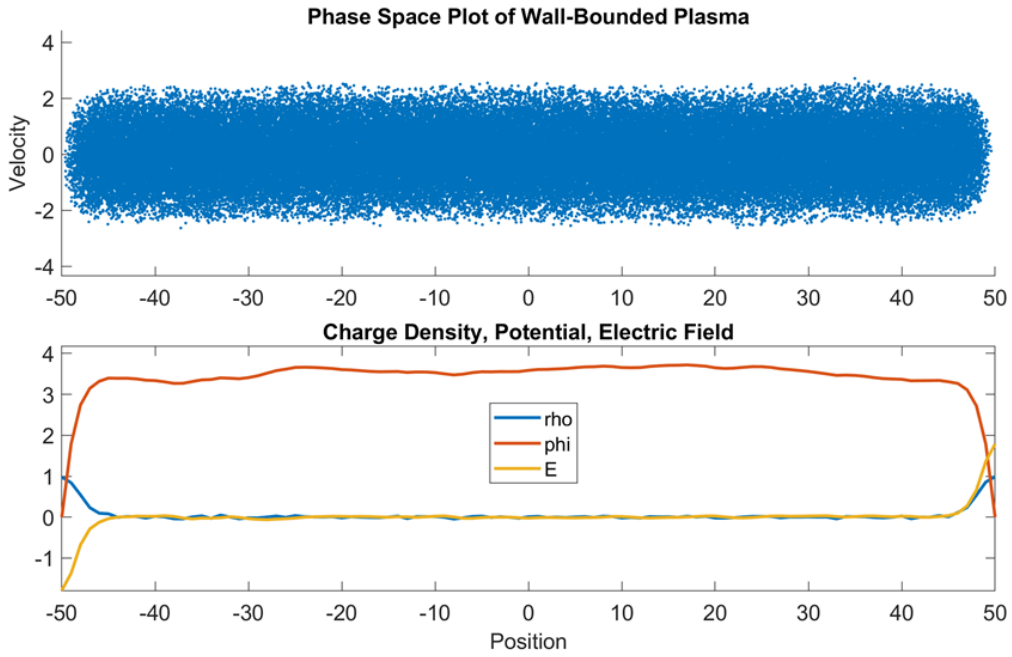


Figure 3: Phase Space positions of particles after 80 periods. Initial conditions: $N = 10^5$, $gp = 101$, $L = 100$.

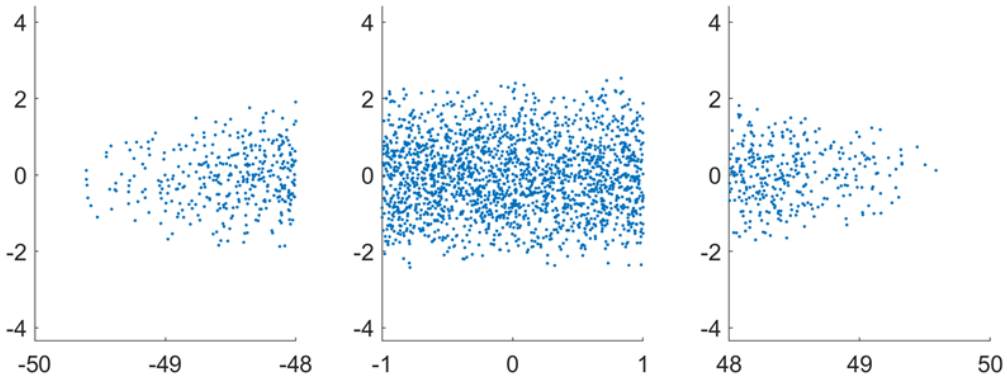


Figure 4: Left-wall, center, and right-wall phase space plots. Taken from the distribution in Figure 3.

3.4 Particle Source

After a period of time, the plasma reaches a steady state, in which no more particles leave the plasma and the sheath is fully developed. Because ions are static, at steady state no electrons are able to penetrate the sheath and reach the wall. In order to observe a realistic energy flux density after steady state is reached, a particle source is introduced. The injection rate of this source is calculated based on the theoretical Bohm flux,

$$\Gamma_x = n v_{se}, \tag{3.26}$$

where the Bohm criterion posits that the fluid velocity of electrons at the sheath edge, v_{se}, is at least equal to the sound speed, c_s:

$$v_{se} \geq \left(\frac{kT_e}{m_i}\right)^{1/2} \equiv c_s. \tag{3.27}$$

Equations (3.26) and (3.27) are used to compute the particle source rate required to maintain Bohm flux. Assume a normalized density of 1, a normalized ion mass of 1836, and a normalized kT_e of 1. The result of (3.26), using (3.27), is $\Gamma_x \approx 1/43$. Our particle weight (W_N) times the number of particles injected per unit time should equal 1/43. For example, consider a domain of 100 Debye lengths containing 10^4 particles. The particle weight is $W_N = 10^{-2}$. Thus, the number of particles injected per unit time is 100/43. Time is normalized by the electron plasma frequency in radians per second,

so this quantity can be thought of as a number per radian. The number per plasma period is found by multiplying by 2π, and is $2\pi(100/43) \approx 14.6$. Taking into account the fact that both left and right walls absorb the Bohm flux, the total source rate (per period) is $4\pi/(43W_N) \approx 29.2$.

To do this, the particle injection rate is converted from number per period to number per time step. For example, with $\Delta t = 0.1$, a source injection rate of $29.2\tau_p^{-1}$ corresponds to about 0.465 particles per time step. This is approximated by injecting one particle per whole integer every time step and carrying over the remainder into the next. Once a whole integer is formed from the accumulated particle fractions, another particle is injected.

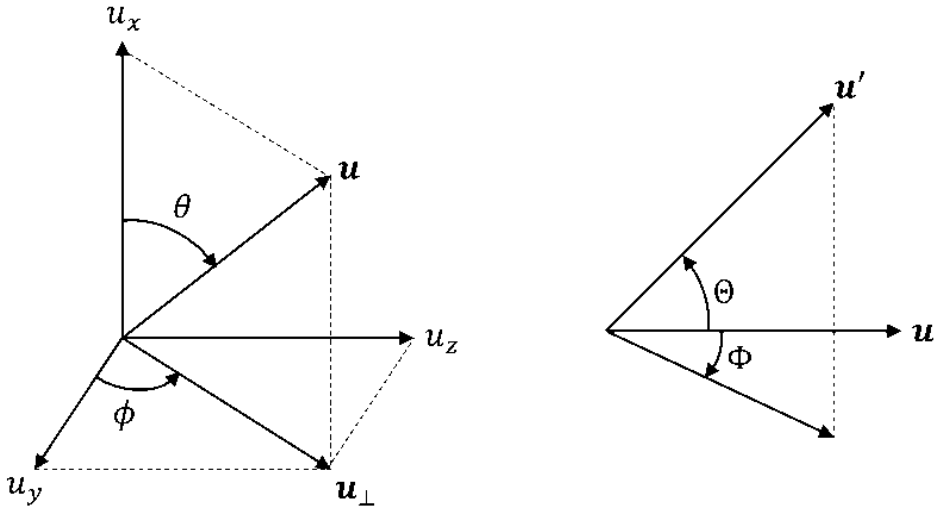
The injected particles are chosen randomly from a Maxwellian distribution in velocity space. They are inserted into the domain across a fifth of the total domain length, centered at $x = 0$. The source electrons are incorporated into the program such that they exhibit the same behavior as those particles that were initialized with the simulation, including Lorentz force weighting, charge density weighting, and Coulomb collisions.

The temperature of the source distribution also requires careful derivation. Since the energy of the plasma leaves at the wall, we want to inject particles with energy flux density $2T_{wall}\Gamma_{src}$. This is equal to the amount of energy leaving at the walls per unit time (see equation (2.9)). For example, if the temperature at the wall is equal to 1, we can solve for the ratio of source energy flux to source injection rate: $h_{src} = 2T_{wall}\Gamma_{src}$. We can see that h_{src} must be 2 times Γ_{src}. Utilizing principles of thermodynamics, we can write the source energy flux density as $h_{src} = \frac{3}{2}T_{src}\Gamma_{src}$. Solving for T_{src}, results in a source temperature of 4/3.

3.5 Coulomb Collisions

Coulomb collisions occur naturally between the constituent particles of a plasma. When temperatures are high, the resulting collisional effects may be ignored in a simulation. However, at lower temperatures, collisional effects play a vital role in the plasma's overall behavior.

The collision operator presented here[3] collides electrons with ions. In three dimensions, an electron trajectory undergoes two rotations upon a collision. These are identified as the scattering angle Θ and the azimuthal angle Φ (Figure 3 (b)). The pre-collision velocities of the electron and ion are $\mathbf{v}_e$ and $\mathbf{v}_i$, respectively, and their post-collision velocities are $\mathbf{v}_e'$ and $\mathbf{v}_i'$.



(a) (b)

Figure 5: (a) In the frame of the laboratory, the relative velocity $\mathbf{u}$ is defined by its spherical components. (b) After a collision, the relative velocity $\mathbf{u}$ is rotated by amounts Θ and Φ (azimuthal).

The post-collision velocities are expressed in terms of their initial velocities, respective masses, and a $\Delta\mathbf{u}$ vector that contains the rotation operator,

$$\mathbf{v}_e' = \mathbf{v}_e + \frac{m_i}{m_e + m_i}\Delta\mathbf{u} \quad \text{and} \quad \mathbf{v}_i' = \mathbf{v}_i + \frac{m_e}{m_e + m_i}\Delta\mathbf{u}. \tag{3.28}$$

where $\Delta\mathbf{u} = \mathbf{u}' - \mathbf{u}$ and $\mathbf{u} = \mathbf{v}_e - \mathbf{v}_i$. The scattering of the relative velocity $\mathbf{u}$ is defined by the operation,

$$\Delta\mathbf{u} = \left(\hat{O}(\Theta, \Phi) - I\right)\mathbf{u}, \tag{3.29}$$

where $\hat{O}$ represents the rotation upon scattering in Θ and Φ,

$$\hat{O} = \begin{pmatrix} cos(\Theta) & -b_y sin(\Theta)cos(\Phi) & -b_z sin(\Theta)cos(\Phi) \\ b_y sin(\Theta)cos(\Phi) & cos(\Theta) & asin(\Theta)sin(\Phi) \\ b_z sin(\Theta)cos(\Phi) & -asin(\Theta)sin(\Phi) & cos(\Theta) \end{pmatrix}, \tag{3.30}$$
$$b_{y,z} = \frac{u_{y,z}}{u_\perp}, \quad a = \frac{u}{u_\perp}, \quad u_\perp = \sqrt{u_y^2 + u_z^2}.$$

The scattering angle Θ is chosen randomly from a Gaussian distribution,

$$P(\Theta) = \frac{\Theta}{\langle\Theta^2\rangle_{\Delta t_c}} exp\left(-\frac{\Theta^2}{2\langle\Theta^2\rangle_{\Delta t_c}}\right), \tag{3.31}$$

where the variance is defined by

$$\langle\Theta^2\rangle_{\Delta t_c} = \frac{e_1 e_2}{2\pi\epsilon_0^2}\frac{n\Delta t_c \ln\Lambda}{\mu^2 u^3}. \tag{3.32}$$

The particle charges are denoted by $e_{1,2}$ and the $\mu = m_e m_i/(m_e + m_i)$ is the reduced mass, while n and $\ln\Lambda$ are the density and Coulomb logarithm, respectively. Solving (3.31) for Θ yields

$$\Theta = \sqrt{-2\langle\Theta^2\rangle_{\Delta t_c} \ln R_1}. \tag{3.33}$$

The azimuthal angle Φ is simply chosen randomly between 0 and 2π,

$$\Phi = 2\pi R_2 \tag{3.34}$$

where R_1 and R_2 are random values between 0 and 1.

The collision time is defined by a limiting condition to assure effective sizing,

$$\nu_c \Delta t_c << 1 \qquad \text{and} \qquad \nu_c = \frac{e_1 e_2}{2\pi\epsilon_0^2}\frac{n\ln\Lambda}{\mu^2 u^3}. \tag{3.35}$$

Since we are assuming ions to be static, a likely choice for T_i would be zero. However, this choice leads to an electron temperature equilibration of $T_e = 0$, effectively killing the plasma. Instead, ions are initialized (for collision purposes) with a Maxwellian velocity distribution in three dimensions at a temperature equal to that of the electrons. This removes temperature equilibration complications, but introduces an energy conservation problem. Since the ions are not allowed to move with their post collision velocities, we find that the ions behave as a small sink of energy. In section 3.6, the implications of this are discussed

Furthermore, electron velocities in the y and z directions are tracked. These additional velocity components are used only in collision computations, but are not modeled using PIC. Unlike in the collision operator, vector components in PIC are independent of each other.

3.6 Energy Balance

Ions are given an initial temperature for the purpose of colliding with electrons. Because of their mass, and to simplify our calculations, we take these ions to be static. This results in a small sink of energy when collisions occur. To account for this, we measure the electron energy loss from collisions and add it to the energy lost at the walls. When taken over a long time interval, this is the total power out. The total power in comes directly from the source.

At the beginning of a simulation, many particles leave quickly, contributing to a large loss of energy, while the power input from the source remains constant. As discussed in section 3.4, one measure of steady state is when the number of particles in your domain is constant. Another common way of representing this period of time is when the percent difference of your power in to power out is constant. All energies will be considered properly accounted for if it is less than 10%. Over nearly 20 periods in steady state, the power input is .0029 and the power output is .0030. This gives a percent difference of 3.4%.

4 Results

The characteristic temperature of the plasma was varied between $T_0 = 0.5eV$ and $T_0 = 2.0eV$ to observe collisional effects. For all tests, the initial number of particles was 5×10^4 and the characteristic density, n_0, was $10^{21}m^{-3}$. Quantitative results are tabulated in Table 1. Figures 6 and 7 display the range of simulated temperature and electron density across the domain for each case. Temperatures were obtained using (2.10) and the densities were calculated by multiplying the number of particles in each cell by W_n. Each were averaged over 20 plasma periods in steady state. The large temperature drop near each wall can be attributed to the non-Maxwellian nature of the velocity distributions in the sheath.

Figure 8 highlights several of the sheath cell distributions. Each subplot was created by concatenating the velocities in a cell over 20 plasma periods into one distribution. The effect of this is that we can observe the statistics of a much more populated plasma. Figure 7(a) shows the distribution of the left-wall cell, which is skewed and missing the higher energies in the right tail. This is expected, as the sheath behaves like a high energy filter. Leftward

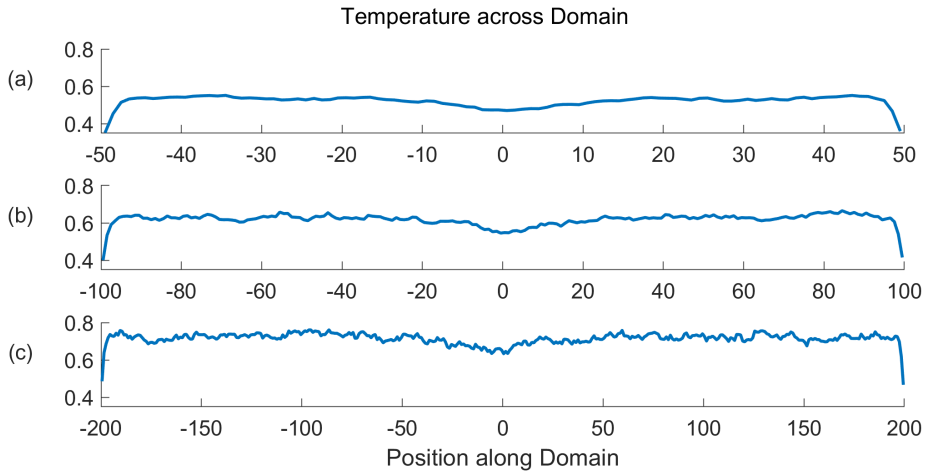


Figure 6: Temperature across the domain for three different characteristic temperatures: (a) $T_0 = 0.5eV$, (b) $T_0 = 1.0eV$, (c) $T_0 = 2.0eV$. The large temperature drop seen at the walls is an artifact due to the distribution not being Maxwellian. The dip in temperature at the middle of each plot is due to the particle source. Each domain length was consecutively increased to account for the increased collision mean free path.

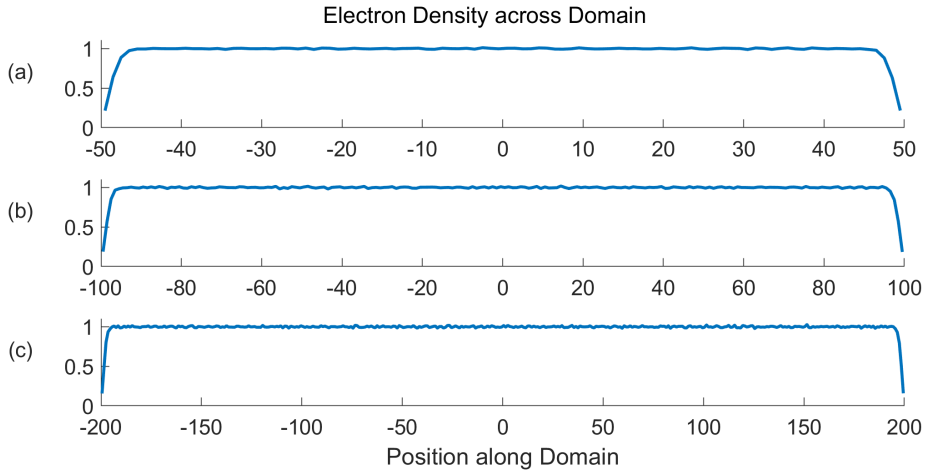


Figure 7: Electron density across the domain for three different characteristic temperatures: (a) $T_0 = 0.5eV$, (b) $T_0 = 1.0eV$, (c) $T_0 = 2.0eV$.

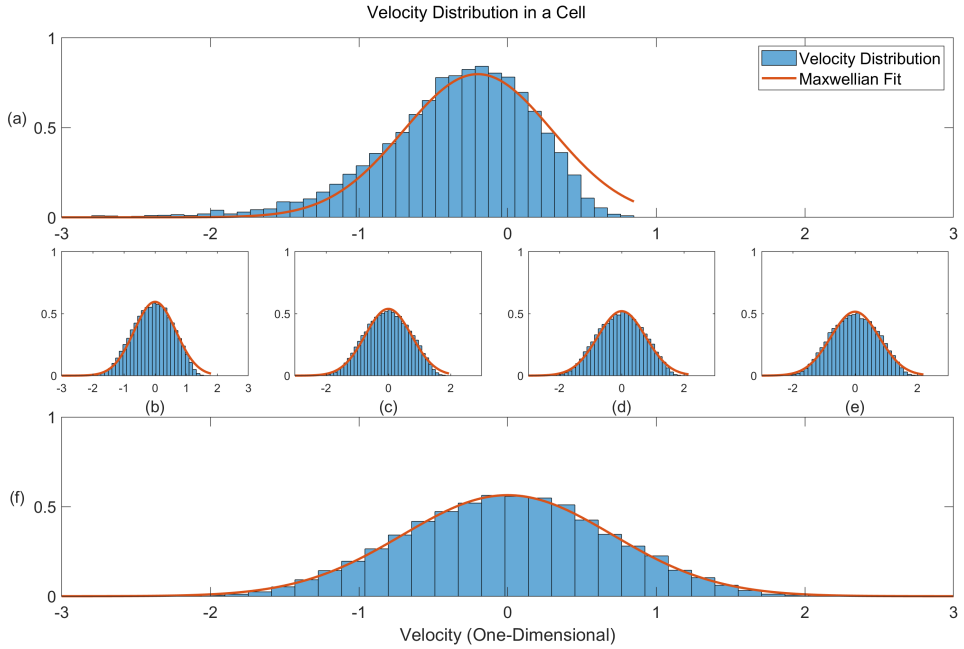


Figure 8: Velocity distributions at cells near the left wall. A corresponding Maxwellian fit is superimposed to highlight the missing tails. (a) Left wall cell, (b-e) consecutive, adjacent cells away from the left wall, (f) domain center cell.

particles with high energy are captured by the left wall, leaving only low energy particles to reverse direction.

This effect is strongest near the wall, and is diminished as we move away, as seen in Figure 8 (b-e). Finally, in the center cell (Figure 8 (f)), we have a proper Maxwellian distribution.

Table 1 shows the results of energy flux calculations and related, important quantities at the sheath edge. The collision mean-free-path, λ_c, indicates the distance each electron travels before it suffers a collision with an ion. The domain length, L, needs to be at least an order of magnitude greater than λ_c in order to adequately resolve collisional effects. The quantities following T_0, λ_c, and L in Table 1 are the result of a long time average at the sheath edge (20 plasma periods in steady state). The sheath edge locations for cases $T_0 = 0.5, 1.0, 2.0$ were 5, 6, and 5 Debye lengths from each wall, respectively.

The energy flux densities, h_{std} and h_k correspond to (2.12) and (2.13),

Table 1: Sheath edge quantities.

T_0(eV)	0.5	1.0	2.0
λ_c	2.8808	8.1481	23.0462
L	100	200	400
h_{std}	0.0615	0.0726	0.0917
h_k	0.0758	0.0852	0.1102
q_k	0.0450	.0488	0.0642
q_f	0.0636	0.1706	0.5872
ϕ	1.2563	1.5706	1.9654
λ_T	125	234	309

respectively. A close and consistent match here indicates that this kinetic model is suitable for predicting accurate sheath edge boundary conditions for fluid codes. In each test case, h_{std} compares closely to h_k in order of magnitude. The one way heat-flux densities calculated using kinetic methods, q_k (2.15), and fluid methods, q_f (2.16), show a much larger percent difference, especially with higher temperatures. A large difference is to be expected, as (2.16) makes fluid assumptions about the plasma in its derivation.

However, we are interested in the temperature gradient, ∇T, across the sheath edge. This value is most appropriately represented by λ_T, the temperature gradient length scale. It is defined as $\lambda_T = T_e/\nabla T$. For $\lambda_T = 125$, this means that the temperature will drop from its value at the sheath edge to 0 over a distance of 125 Debye lengths. Put into real length scales (using characteristic values $T_0 = 0.5eV$ and $n_0 = 10^{21}m^{-3}$), 125 Debye lengths is equivalent to about $20\mu m$. In tokamak modeling, for example, the simulated domain length is on the order of several meters, making a scale of this magnitude difficult to resolve. With three test cases, the only observable trend is that λ_T seems to increase with temperature. At higher temperatures, $\kappa_{\|}$ becomes large and a small temperature gradient can maintain the total energy flux (the other portion comes from the kinetic and internal energy fluxes of (2.13)). Since smaller ∇T corresponds to larger λ_T, we do in fact observe larger temperature gradients at low plasma temperatures. An improved grid resolution and more temperature test cases can help to produce a more refined trend.

However, in Figure 6, the plots show that the temperature gradient varies significantly near the sheath edge ($\sim$5 Debye lengths into the plasma). Future

work could probe for a deeper understanding of the heat flux composition. For example, one might want to know if the direction of q_k can be in the positive ∇T direction, carrying heat "uphill".

Additionally, as seen in Figure 3, the electric potential at the sheath edge is about 3 without collisions. Lower temperatures (more collisions) appear to reduce the potential in the sheath edge in steady state (although, the simulated temperature is also reduced such that the potential is roughly three times this temperature at the sheath edge). We expect that with higher T_0, the sheath edge potential will approach 3, as collisional effects become less pronounced.

5 Conclusion

By utilizing Coulomb collisions and a particle source, the kinetic model presented here has shown that it can reliably produce energy flux boundary conditions for fluid codes at low temperatures. This is evidenced by the close relationship between h_{std} and h_k in all three cases observed. Significantly more work would be required to expound on what those boundary conditions would be in practice.

The quantities explored in Table 1 are by no means an exhaustive list of what can be learned from the plasma-wall interface. Additional work could entail: 1) Improving the spatial grid resolution; the first cell near each wall experiences a large potential drop. Improving the grid resolution there would allow for a better picture of the potential between those two grid points. 2) Sampling a greater range of plasma temperatures; this would help inform trends in the data. 3) Implementing electron-electron collisions, as well as collisions with neutrals; this will improve the accuracy of the simulation results. Electron-ion collisions helped bring the results to an appropriate order of magnitude, but the complete collision picture can refine them even more. 4) Allowing ion motion; static ions helped to simplify the collision operator and PIC code, but may have affected the sheath physics. Allowing ion dynamics may reveal different, more realistic, sheath behavior.

The computational program detailed in this project may be used as a tool for plasma research in modeling kinetic, low temperature plasmas. This energy flux study was a preliminary look at the physics of the plasma-wall interface using the program.

References

[1] P.C. Stangeby, *The Plasma Boundary of Magnetic Fusion Devices*. Bristol: Institute Of Physics Pub., 61-98 (2000).

[2] C. Birdsall and A. Langdon, *Plasma Physics Via Computer Simulation*. Bristol: Institute of Physics Pub. (2005).

[3] D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna, "The Particle-In-Cell Method", *Contrib. Plasma Phys.* **47**, No. 8-9, 563-594 (2007).