Development of the Starfish Plasma Simulation Code and Update on Multiscale Modeling of Hall Thrusters

Lubos Brieda*
Particle In Cell Consulting LLC, Falls Church, VA 22046
Michael Keidar†
The George Washington University, Washington, D.C.

This paper summarizes recent development on a multiscale approach for analyzing Hall thrusters. The approach is based on the thermalized potential model utilized in codes such as HPHall. However, instead of relying on analytical expressions for the cross-field electron mobility, the mobility is computed self-consistently with a kinetic code. In addition, a two dimensional code is used to study the ion dynamics in the sheath. In this update, we discuss the numerical model used by these two modules. Integration of these codes into a single end-user product required development of a replacement axisymmetric solver. The new code, a generalized plasma solver Starfish, is also described. The paper then presents latest results. These include study of diffusion processes which indicates that synergetic effects play an important role in electron transport. We also investigate the sheath in the presence of an inclined magnetic field, and find that for large angles, the sheath collapses, and ions are accelerated away from the wall.

I. Introduction

Hall thrusters are spacecraft propulsion devices that utilize an applied magnetic field to create thrust. Despite over 40 years of flight heritage, the community still lacks a tool capable of predictively modeling these devices. Contributing to this difficulty is the presence of multiple spatial scales at which physics of importance occurs. Example of one such effect is the lack of a general model for describing diffusion of electrons across magnetic field lines. The classical model for electron cross-field mobility is solely driven by collisions of electrons with heavy ions and atoms. The model does not take into account walls, non-steady discharge, and non-uniformities in temperature. This model is also insufficient to reproduce the anode currents observed experimentally. Additional models for capturing some of the non-classical processes have been incorporated into existing codes with somewhat mixed results. The reason for the discrepancy arises from the fact that electron transport is inherently a kinetic process occurying on the spatial scale of electron motion. The details of importance are lost in the transition to the macroscopic world described by the mobility models.

Due to numerical limitations, it is not feasible to model real-sized Hall thrusters while resolving details of electron motion without resorting to non-physical adjustments of electron mass or plasma permittivity. Instead, the common approach for modeling these devices is based on the thermalize potential model (TPM) outlined by Morozov,[1] and later implemented by Fife in HPHall.[2] These codes assume that since in a magnetized plasma a $D_{\parallel} >> D_{\perp}$ anisotropy exists in the diffusion coefficient, the magnetic field lines become lines of constant temperature T_e and thermalized potential ϕ^*. The dimensionality of the problem is thus reduced to one-dimension, since only the characteristic values need to be computed. These values are functions of the magnetic field line, $T_e = T_(\lambda)$ and $phi^* = \phi^*(\lambda)$. HPHall utilizes two sets of computational meshes. Ion and neutral dynamics is computed on an elliptic quadrilater mesh, with both species treated by the particle-in-cell method. The kinetic treatment allows HPHall resolve the ion velocity distribution

*President, Member AIAA, lubos.brieda@particleincell.com
†Associate Professor, Department of Mechanical and Aerospace Engineering, Member AIAA

function self-consistently. The electron conservation equations are solved on a "lambda mesh", in which the radial grid lines correspond to the magnetic stream lines. The cross-field electron velocity appears in the momentum and energy equation and is obtained from the Ohm's law,

$$u_{e,\hat{n}} = -\mu_{e,\perp}\left(E_{\hat{n}} + \frac{1}{en_e}\frac{\partial p_e}{\partial \hat{n}}\right) \tag{1}$$

Here, $\mu_\perp$ is the electron mobility across the magnetic field discussed previously. The model used by the original HPHall is

$$\mu_{e,\perp} = \frac{\mu}{\beta_e^2} + K_B\frac{1}{16B} \tag{2}$$

where $\mu = |q|/m\nu_{ea}$ is the classical mobility, $\beta = \omega_c/\nu_{en}$ is the Hall parameter, and K_B is the Bohm coefficient, controlling the strength of the anomalous Bohm mobility (the second term in equation 2). What can be immediately noted is the absence of kinetic effects in this relationship. The Bohm term is independent of the electron or ion population and is a simple function of the magnetic field strength. Electron details appear only in the form of the electron-atom collision frequency ν_{ea}. In the recent years, the model had been extended to include near-wall conductivity,[3,4] shear-based transport,[5] and zones of low and high mobility.[6] These models suffer from the same limitations as the original formulation. To illustrate this, let's consider the near wall conductivity (NWC). The additional electron current due to wall interactions is assumed to scale with $\Gamma_{e,s} = \gamma\Gamma_{e,p}$, where the two fluxes correspond to the secondary (reflected and true emitted secondaries) and primary (incident) electrons. The secondary electron emission coefficient scales with electron temperature, $\gamma = \gamma(T_e)$, however only a single characteristic temperature is considered. As has been demonstrated previously,[7,8] electron temperature in Hall thrusters deviates from Maxwellian and is anisotropic, $T_{e,\perp} \neq T_{e,\parallel}$. Bulk of electrons are screened from the wall by a potential gradient with only the energetic particles being able to penetrate to the wall. The yield of secondary electrons is governed by the high energy tail of the electron distribution function, and not an average temperature.

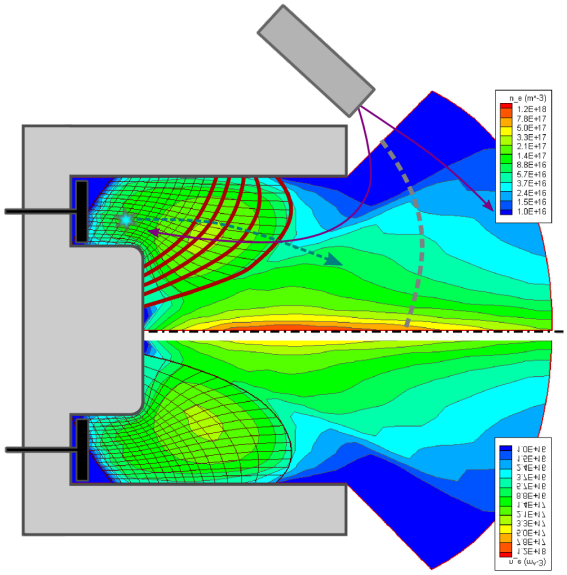


Figure 1. The setup for a typical Hall thruster simulation based on the thermalized potential approach.

Figure 1 graphically illustrates the thermalized potential model as applied to the Princeton Cylindrical Hall Thruster.[9] This figure also shows several important components of a Hall thruster, including the anode and the cathode. A representative λ mesh is superimposed over the HPHall simulation. It should be noted, that the λ mesh constructed by HPHall has a variable number of partitions along each field line and thus does not form an actual topological domain. The illustrative mesh used in this figure uses the average number of radial partitions. The electron energy equation is solved only on this λ mesh. The output from the solver is $T_e(\lambda)$ and $\phi^*(\lambda)$, where the second term is known as the thermalized potential. Just like temperature, ϕ^* is constant for each field line. The radial variation in plasma potential is obtained from $\phi = \phi^* + kT_e/e\ln(n_e)$

assuming quasi-neutrality everywhere, $n_e = n_i$. Constant potential is applied upstream of the mesh, while a linear interpolation to ground is used downstream.

A. Multiscale Formulation

The thermalized potential model (TPM) is attractive for modeling Hall thrusters. It is computationally efficient since only several hundred computational cells are required (typical HPHall simulations utilize domains with approximately 50×30 cells). In the PIC method, the number of required particles scales directly with the number of cells, since several particles are required per cell to characterize the local VDF. Also, since the simulation progresses on the time scale of ions, ony several tens of thousands steps are required to resolve the breathing mode oscillations. A fully kinetic simulation, on the other hand, requires tens of thousands simulation cells to resolve the local Debye length, significantly higher particle counts, and time steps on the scale of electron cyclotron frequency. Even for a miniaturized thruster, this calls for the use of a large supercomputer.

The TPM model is also physically sound. On the spatial scale of ions, electrons are seen to respond instnteneously and as such, it is reasonable to describe them with a fluid model. The approach however suffers from two shortcomings. First, the fluid tretment of electrons coupled with the simplified analytical mobility results in the aforementioned under-prediction in anode current. Secondly, the universal quasi-neutrality assumption prevents thecode from resolving the near wall ion dynamics. The direct manifestation of this shortcoming is the inablity of ions lost to the wall to reach the Bohm velocity, which then necessitates the need for additional post-processing on the impact ion velocity.[10]

In this paper we discuss an alternative approach for modeling Hall thrusters based on a multi-scale formulation. Our formulation attempts to address these two shortcomings while retaining the computational efficiency of the TPM. First, instead of utilizing analytical models for $\mu_\perp$, our approach computes the cross-field mobility self-consistently by considering only the first principle laws. The calculation is performed by treating the electrons as kinetic particles and simulating their orbits about the magnetic field line. Mobility is obtained from the speed at which the guiding center diffuses, $\mu_\perp = v_{gc}E_\perp$. What distinguishes our approach from a fully-kinetic thruster simulation is that only a subset of the domain is considered. The thermalized potential solver requires the value of $\mu_\perp$ only at the spatial locations corresponding to grid nodes of the λ mesh. Since in a Hall thruster the electrons are magnetized, each magnetic field line can be thought of as an independent domain. Coupling between the magnetic field line domain and the global discharge is provided by $E_\perp$, and the heavy particle densities and velocities, n_a, n_i, and u_i. As such, we can obtain mobility self-consistently by considering only a small number of magnetic field lines. The number of λ grid lines in a typical HPHall simulation is on the order of 20. By utilizing modern processor multi-core architecture, we can simulate multiple field lines in parallel. Further reduction in computational effort is possible by reducing the number of analyzed locations and simply interpolating values onto the intermediary λ lines. The second shortcoming is related to the plasma-wall interface. In order to resolve ion wall flux and, more generally, the ion motion in the sheath, the code must take into account the deviation from quasineutrality in the sheath. The Poisson's equation, $\nabla^2\phi = -\rho/\epsilon_0$, must thus be solved. This requires a mesh capable of resolving the Debye lenght λ_D. Since this solver only address the near-wall region, the spatial dimensions are reduced, and the solution can be obtained on a standard workstation in the matter of minutes. We use this model to correct the wall fluxes from the TPM code. Since ions impacting the wall recombine are are effectively removed from the simulation, the coupling is one-way only.

Figure 2 presents a graphical representation of our multiscale approach. We first use a TPM solver and the standard mobility model given in Equation 2 to obtain the initial solution. Since Hall thruster discharges reach only an oscillatory-type steady state, this initial solution corresponds to properties averaged over several breathing mode oscillations. We next use a kinetic code Lynx to compute the mobility along the required magnetic field lines. The self-consistent mobility is then used to obtain a new thruster solution. The process repeats until convergence, $\partial\mu_\perp/\partial t \leq \epsilon_{tol}$. The resulting solution is subsequently used to establish the bulk-plasma boundary for 2D sheath code used to study ion motion and erosion in the sheath. In addition, ions exiting the thruster in the TPM simulation can be sampled to obtain a discretized source model for a electric propulsion plume modeling analysis. This step, which is not addressed in this paper, is described in more detail in 8.

The general concept of the multiscale approach has been reported previously in 11,12. In this paper we discuss the latest developments. We first provide details of the Lynx kinetic code as parts of it had gone significant changes from the initial version. Our prior proof-of-concept studies utilized HPHall as the TMP

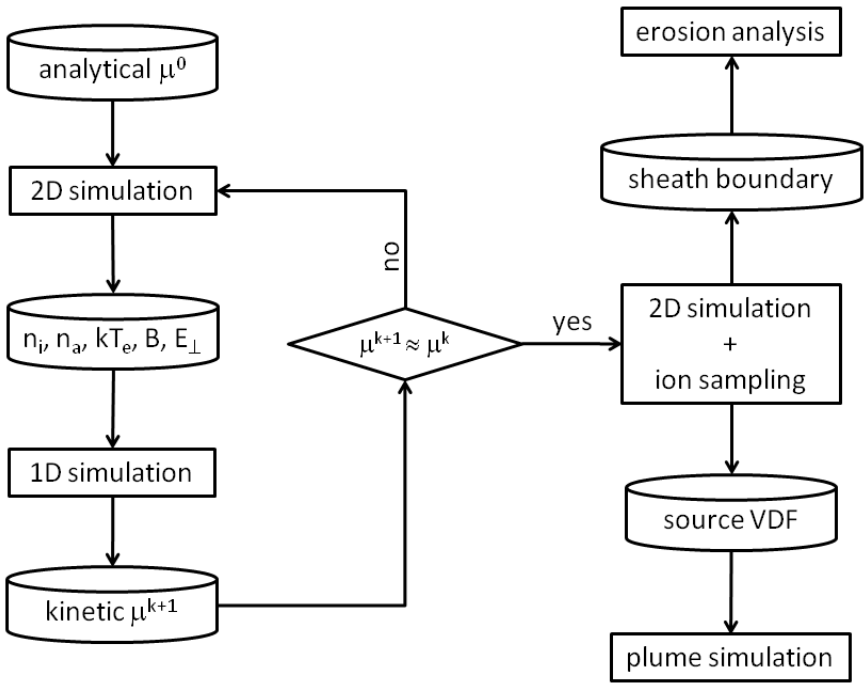


Figure 2. Schematic of the multiscale approach

solver. As detailed in the following paragraphs, a number of issues lead to our decision to develop a new solver. We have began work on a general 2D plasma / rarefied gas solver Starfish. This code is described next. Subsequently, we describe details of the recently developed 2D sheath code for analyzing ion wall flux as well as some recent models for the plasma-wall interface. The paper then continues with results. We first present findings from a mobility study indicating that synergystic effects play an important role in transport. We also discuss the near-wall ion motion for a thruster with a steeply inclined magnetic field lines. The paper is concluded with example result obtained with the Starfish code.

II. Kinetic Modeling of Electron Transport

This section provides a general overview of the kinetic code Lynx used to kinetically compute electron transport across magnetic field lines. The code is described in more detail in 8. Lynx uses a hybrid version of the particle-in-cell (PIC) technique. In this formulation, only electrons are simulated. Ions and neutrals are assumed to be stationary and are described by a fixed background density. In a Hall thruster, ions are not magnetized, and are continuously streaming across the field line. Keeping ions stationary imposes the assumption that mobility develops rapidly in response to a change in global discharge parameters. This is believed to be a valid assumption, since in the frame of reference of ions, electrons respond instantaneously to disturbances due to their light mass.

The input for a Lynx simulation is an HPHall-like λ mesh containing the global discharge parameters, n_i, n_a, $E_\perp$, B, T_e, and $u_{i,\perp}$. The code assumes that n_i is the total charge-averaged ion density, i.e., ion charge density $\rho_i = en_i$. The main handler then launches an individual simulation for each radial grid line of the input λ mesh. The simulations are performed in parallel utilizing Java's native support for multi-threading. Each 1D simulation begins by the code creating electron particles following the prescribed ion density, $n_e = n_i$ everywhere. The parallel component of electric field $E_\parallel$ is determined self-consistently by solving Poisson's equation, $\nabla^2\phi = -\rho/\epsilon_0$ at each time step. Electrons are advanced according to the leapfrog method. The code initially runs in a sheath-forming mode in which electrons impacting the walls are absorbed and collisions are ignored. This mode continues until a well-formed sheath $j = 0$ forms at both walls. Since the high-energy electrons will be preferentially lost during this sheath formation stage, the code subsequently resamples the electron population to regenerate the high-energy tail.

The simulations then continues in a normal mode in which a detailed wall and collision models are employed. Mobility is recovered from the speed with which electron guiding centers diffuse. Three primary transport mechanisms are considered: particle collisions, wall interaction, and electric field instabilities. Collisions are modeled with a Monte Carlo approach and several collisional processes, such as momentum transfer (scatter), Coulomb collisions, ionization, and excitation, are included. Wall interactions are modeled following the approach of Sydorenko.[13] Support for modeling azimuthal waves is also included, however, this mechanism is generally not used. Computational times are on the order of minutes for a typical simulation consisting of 100,000 particles and 20,000 time steps. Vast majority of the computational effort is spent averaging the results, since the steady state is typically reached within the first 1000 time steps.

A. Computational Domain

The computational domain for each 1D analysis consists of a curved magnetic field line connecting the inner and the outer thruster wall. Along this domain, the coordinate system consists of two components: the distance s along the field line, and a coordinate system $x-y$ in the plane normal to the line. The distance is measured from the inner wall. The other two coordinate directions, $\hat{\vec{e}}_x$ and $\hat{\vec{e}}_y$ are normal to $\hat{\vec{e}}_s$. These coordinates are defined such that $\hat{\vec{e}}_x$ lies in the $r-z$ plane of the thruster, while $\hat{\vec{e}}_y$ lies in the $r-\theta$ plane. In the code, it is assumed that electron orbital planes rotate automatically with the magnetic field line. As such, the actual profile of the field line is not required in the particle push. The magnetic field geometry is however used to compute the field gradient, $\partial B/\partial s$ by differentiating the supplied $B = B(s)$ magnetic field profile. The incidence angle at the inner and the outer wall is also of importance when considering injection of secondary electrons at the wall.

The field line is discretized into a number of equidistant computational segments (one-dimensional cells) in accordance with the particle-in-cell (PIC) method. The length of each segment is Δs. This cell spacing is set such that $\lambda_{D,min} = h\Delta s$, where $\lambda_{D,min} = \sqrt{\epsilon_0 k T_e / n_{p,max} e^2}$ is the minimum Debye length along the entire simulation domain, and h is a user provided parameter. Typically, $h = 2$ is used, resulting in two cells per Debye length. Debye length is calculated using the constant initial electron temperature, which is assumed to be constant along the field line $\partial k T_e/\partial s = 0$ and isotropic, $kT_{e,\perp} = kT_{e,\parallel}$. It should be noted that prior analysis of Sydorenko[7] as well as our own work indicate that electron distribution function in Hall thrusters becomes anisotropic due to the presence of cold secondary electrons. In the simulation, the kinetic EEDF is characterized, however this value is used only for diagnostic purposes. New electrons are sampled from the initial distribution.

B. Potential Solver

Lynx solves the electric field in the parallel direction, $E_\parallel$ self-consistently from the Poisson's equation, $\nabla \cdot \vec{E} = -\nabla^2 \phi = \rho/\epsilon_0$. The discretized form of this equation is determined from the finite volume approach. We can define a control volume around each simulation node and perform the integration

$$\begin{aligned} \int_V \nabla \cdot \nabla \phi dV &= \int_V -\frac{\rho}{\epsilon_0} dV \\ \int_S \nabla \phi \cdot \hat{n} dA \cong \sum_{i=1}^{4} \nabla \phi \cdot \hat{n} dA &= -\frac{\rho}{\epsilon_0} r \Delta x \Delta y \Delta s \\ -\nabla \phi_1 r_1 + \nabla \phi_3 r_3 &= -\frac{\rho}{\epsilon_0} r \Delta s \end{aligned} \tag{3}$$

where $r = f(s)$ and $\nabla\phi_2$ and $\nabla\phi_4$ are zero since for the one dimensional case we have $\partial\phi/\partial x = 0$. Axisymmetry dictates $\partial\phi/\partial y = 0$. Here the indexes correspond to the control volume face, starting with the bottom face and continuing around the control volume counter-clockwise. We next need an expression for the two derivatives $\nabla\phi_1$ and $\nabla\phi_3$. These are obtained using the scalar form of the divergence theorem, $\int_S \nabla\phi dA = \oint \phi \hat{n} dl$. For our formulation this yields the standard finite difference derivate, $\nabla\phi_1 = (\phi_j - \phi_{j-1})/\Delta s \vec{\hat{e}}_s$ and $\nabla\phi_3 = (\phi_{j+1} - \phi_j)/\Delta s \vec{\hat{e}}_s$. Substituting these expression into Eq. 3 gives

$$\frac{\phi_{j+1} - \phi_j}{\Delta^2 s} \frac{r_{j+1/2}}{r_j} - \frac{\phi_j - \phi_{j-1}}{\Delta^2 s} \frac{r_{j-1/2}}{r_j} = -\frac{\rho}{\epsilon_0} \tag{4}$$

For the case of a fully radial magnetic field line where $s \equiv r$ and $\Delta s \equiv \Delta r$, this expression simplifies to

$$\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta^2 s} + \frac{1}{r}\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta r} = -\frac{\rho}{\epsilon_0} \tag{5}$$

which is the standard discretization of $\nabla_r^2 = \partial^2/\partial r^2 + (1/r)\partial/\partial r$. The electric field $\vec{E} = -\nabla\phi$ is obtained by numerically differentiating the potential, $E = (\phi_{j-1} - \phi_{j+1})/2\Delta s$. This differentiation is independent of the geometry. A one sided model is used along the walls.

C. Boundary Conditions

The expression given in Eq. 4 holds inside the simulation domain. This expression needs to be modified along the walls to take into account the boundaries. Two types of boundary conditions are implemented in the code: conductor and dielectric walls. Dielectric walls are of interest for the SPT-type thrusters. The discretization along the inner wall can be determined by considering a small volume elements centered at the $j = 0$ grid node. From Gauss' law, we have

$$\int_V \nabla \cdot \vec{E} ds\, dA = \frac{Q}{\epsilon_0} \tag{6}$$

where Q is the total charge enclosed by the volume. Applying the divergence theorem and considering a domain with the field varying in only a single direction, we obtain $(E_{1/2} - E_{-1/2})dA = (\sigma + 0.5\Delta s\rho)dA$. This formulation assumes that $dA_0 = dA_{1/2} = dA_{-1/2} = dA$, i.e. a planar geometry. The variation in dA is not considered in the code due to the small Δs spacing between nodes and hence a negligible difference in r. The area element can then be eliminated, giving the boundary condition

$$E_{1/2} = \frac{\phi_0 - \phi_1}{\Delta s} = \sigma + \frac{1}{2}\rho\Delta s \tag{7}$$

The internal electric field, $E_{-1/2}$ is assumed to be zero inside the dielectric. A more detailed treatment, including the correction due to the varying r is found in.[14] Since a potential solution is unique only up to a fixed constant, at least a single Dirichlet boundary condition must be applied to define the problem. This condition is applied by fixing the potential on the outer wall to 0 V, following the formulation in[14] and.[13] It should be noted that this approach appears to neglect the surface charge accumulated on the outer wall. The outer surface charge is included by the condition of charge neutrality, $\int \rho dV + \int(\sigma_0 + \sigma_{nj-1})dA = 0$.

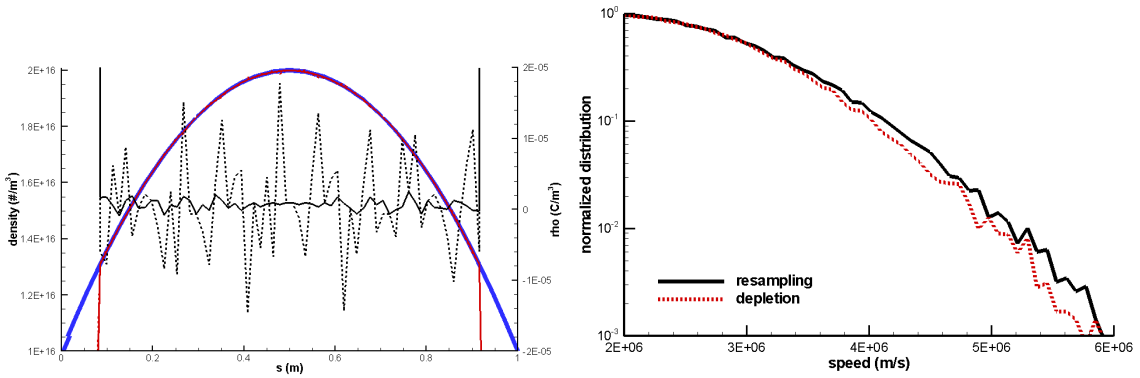
The resulting system of equations produces a tri-diagonal matrix that can be easily solved using the direct Thomas Algorithm. Electric field is then retrieved from $\vec{E} = -\nabla\phi$, or $E_s = (\phi_{j-1} - \phi_{j+1})/(2\Delta s)$. Tthis method produces just one of three electric field components acting on the particle. The other two components are $E_\perp$ and E_θ. The first of these comes directly from the axisymmetric solution and is driven by the potential drop along the thruster channel. The second, E_θ, is the electric field in the azimuthal direction. Assuming the ideal axisymmetric case, this field is zero. However, azimuthal waves and additional oscillations oscillations[15] have been previously observed experimentally. These waves are believed to be an important contributor to electron transport. To investigate their role, Lynx includes an analytical harmonic oscillator that can be used to modulate the azimuthal electric field. The magnitude is given by $E_q = E_0 + A\cos(-\omega t + \varphi)$. This modulation is generally not used, however it is a useful tool for characterizing the influence of electric field fluctuations on electron transport.

D. Particle Loading

The code contains two subroutines for generating particles: (a) quiet start and (b) random injection. The quiet start method is used to initially populate the electron species. This method creates particles at node locations such that $\rho = n_i - n_e \sim 0$ (limited by the statistical weight of particles). If total electron and ion charges $Q_e = \int_s en_e ds$ and $Q_i = \int_s n_i ds$ are initially equal $Q_e = Q_i$ this loading will result in $\rho \sim 0$ everywhere. The initial potential energy of such a system $\int_s \rho\phi \sim 0$ and the energy is determined by electron temperature alone. The simulation begins by loading such a quasineutral system. Lynx uses a constant specific weight (the number of real particles represented by a single simulation macroparticle). Although other researchers[16] indicate that a variable specific weight should be used to represent the high energy tail of the velocity distribution function, such an approach was not used at present due to it imposing its own

difficulties. Namely, variable weight models typically rely on methods for combining and splitting particles. Care must be taken when combining two particles of non-equal velocities to assure that numerical changes in the VDF are not accidentally included. For this reason, a constant specific weight model was used, with implementation of the variable weight model identified as a possible future enhancement. The ability of this model to resolve the high energy tail is further discussed in the next chapter.

The specific weight of each electron is determined by integrating the ion density along the field line to determine the total number of ions in the 1D domain, $N_i = \int_s n_i dV$. The specific weight is then set such that a user-defined number of macroparticles M_0 will represent this count, $w_e = N_i/M_0$. The quiet start routine begins loading this number of particles from center of the magnetic field line. Particles are first loaded at the location of the center node(s), and the loading continues towards the boundaries until the specified number of particles is loaded. The number of particles to load at each node is given by $p = n_{i,j}\Delta V/w_e$, where $n_{i,j}$ is the ion density at the $j-$th node and ΔV is the cell volume. $0.5\Delta V$ is used along the boundaries. Since the particle count p will in general be non-integral, the scheme keeps a track of fractional particles and an additional particle is loaded whenever the fractional count reaches unity.



(a) Electron and ion densities, and charge density for $Q_e = 0.9Q_i$

(b) Logarithmic plot of high energy velocity tail

Figure 3. Particle loading

The importance of this scheme is its ability to load a sheath-like electron distribution. In a wall-bounded plasma, $Q_e < Q_i$ due to a faster initial depletion of electrons in the sheath. The ratio of Q_e/Q_i at steady state is not known *a priori* and must be determined self-consistently. The approach used by the code is to first load an electron population with $Q_e = Q_i$ and allow the simulation to establish a steady state marked by zero wall current. The charge ratio then becomes N_e/N_i, assuming $Z_i = 1$. The number of electrons at steady state is recorded and subsequently used to reload the electron population. Reloading is accomplished by first destroying all particles and then injecting a new population with a reduced electron count. This reloading allows the simulation to replenish the high velocity tail of the electron EDF that is initially lost to the wall. Since electrons are loaded from the center out, particle loading with $Q_e < Q_i$ will result in quasineutrality in the bulk, and a near-wall region where $Q_e = 0$. This is demonstrated in Figure 3. This picture shows electron loading for an analytical quadratic ion density variation. The blue curve shows the ion density, while the red curve shows the neutral density. In this loading, $Q_e = 0.9Q_i$ was used. The black curves show the charge density $\rho = e(n_i - n_e)$. The effect of particle count is shown by the two black curves. Increasing the count reduces the noise in ρ. However, for both cases $\rho \sim 0$ in the bulk. It should be noted that this loading scheme assumes that the sheaths at the inner and the outer wall are symmetric.

Related to particle loading is the internal characterization of a fully developed plasma sheath. The total net current flux to the walls is $\Gamma_{tot} = \Gamma_{outer} + \Gamma_{inner}$, where each flux consists of ion, electron, and secondary electron fluxes, $\Gamma = \Gamma_i - \Gamma_e + \Gamma_s$. The ion flux is obtained by assuming that ions enter the wall with Bohm velocity, $v_b = \sqrt{kT_e/m_i}$. Currently, the actual ion velocity in the near wall region obtained from the 2D code is not taken into account. The sheath is established once $\Gamma_{net} >= 0$.

The above described method is used to load the initial electron population. It is not used during the normal simulation to replenish electrons lost to domain boundaries. New particles are sampled at a random spatial location, with the probability following the prescribed ion density. This scheme works by selecting a random position along the field line, $s = RL$ where R is a random number and L is the domain length. The

probability of a particle being found at this position is obtained from $P = N_i(s)/\sum N_i$. This probability is compared to another random number R_2. If $P \geq R_2$, a particle is placed at this location. Otherwise, the process continues.

E. Loading a Prescribed VDF in Magnetized Plasmas

The particle sampling routines described above select the initial s position of the particle. The initial velocity components $v_\perp$ and v_θ are obtained by sampling the Maxwellian distribution function at the specified temperature, kT_e. Isotropic distribution is used in particle loading. In the parallel direction, the velocity component is given by the standard one-dimensional Maxwellian distribution function

$$f_M = (m/2\pi kT)^{1/2} \exp\left(-v^2/v_{th}^2\right) \tag{8}$$

Particles are sampled from this distribution by selecting a random velocity in the $-6\sigma : +6\sigma$ range and evaluating the probability $P = \exp(-v^2/v_{th}^2)$. The process continues until $P \geq R$. In the perpendicular direction, the velocity component is obtained by a combination of two independently sampled one-dimensional distributions, $v_\perp = \sqrt{f_{M1}^2 + f_{M1}^2}$. Once the desired $v_\parallel$ and $v_\perp$ velocity components are known, the actual particle is created. As will be discussed next, Lynx uses the position of guiding centers to determine mobility. As such, it is imperative that particles are correctly placed on an orbit centered at $x_{gc} = 0$. In addition, the loading function must assure that the correct velocity distribution function is retrieved when particle velocities are averaged over an orbital period. Both of these requirements can be satisfied by placing the particle at a random phase location along the analytical orbit. The velocity components of a particle drifting in an $(E_\perp \vec{e}_x) \times (B\vec{e}_s)$ field are given by[17]

$$v_x = v_\perp \cos(\omega_c t) \tag{9}$$

$$v_y = - \pm v_\perp \sin(\omega_c t) - \frac{E_\perp}{B} \tag{10}$$

These expressions can be integrated to obtain

$$x = r_L \sin(\omega_c t) \tag{11}$$

$$y = \pm r_L \cos(\omega_c t) \tag{12}$$

where $r_L = v_\perp/\omega_c$ is the Larmor radius and $\pm$ follows the particle charge. The $E_\perp/B$ component is ignored in the position, since the position in the azimuthal direction is not relevant.

F. Particle Motion

Motion of charged particles in an electromagnetic field is governed by the Lorentz force,

$$\vec{F} = m\frac{d\vec{v}}{dt} = q\left(\vec{E} + \vec{v} \times \vec{B}\right) \tag{13}$$

Care must be taken when integrating this equation since velocity appears on both sides of the equation. The obvious integration by the forward method, $\vec{v}^{n+1/2} = \vec{v}^{n-1/2} + q/m(\vec{E} + \vec{v} \times \vec{B})\Delta t$ will result in a non-physical energy gain and the particle describing a spiral orbit. Lynx follows the method of Boris[18] which described an elegant alternative to the matrix inversion required by an implicit solver. This method splits the push into a half acceleration, two rotation steps, and another half acceleration.

In the presence of a converging magnetic field, a mirror force, $F_M = -\mu_M(\partial B_s/\partial s)$ retards the motion of particles entering a region of an increasing magnetic field strength. The μ_M term in the above equation is the magnetic moment. It is given by $\mu_M = (1/2)mv_\perp^2/B$. If the mirror is particularly strong, the particle will reach a position where $v_\parallel = 0$ and the particle will be reflected. In Hall thrusters, mirror arises in thrusters utilizing cusped field configurations, or near the centerpole of the CHT due to the geometry of the device. The total energy of a particle in a magnetic field must be conserved, indicating that as the parallel component of velocity decreases, the tangential velocity component must increase to keep the total kinetic energy constant,

$$\frac{d}{dt}\left(\frac{1}{2}mv_\parallel^2 + \frac{1}{2}mv_\perp^2\right) = 0 \tag{14}$$

The Boris algorithm by itself does not resolve this effect. The reason is due to the fact that this relationship arises from the conservation of energy, while the Boris method is integrating the momentum equation. Instead it must be superimposed on the integrator. In the code, the $dv_M = (1/m)F_M\Delta t$ parallel component is first computed. The new perpendicular velocity is computed from $v_\perp^2 = v_0^2 - (v_\parallel + dv_M)^2$. The x and y velocity components are then rescaled accordingly. Particle position is updated using the Leapfrog method,

$$\vec{x}^{n+1} = \vec{x}^n + \vec{v}^{n+1/2}\Delta t \tag{15}$$

where the velocity is integrated from $\vec{v}^{n-1/2}$ to $\vec{v}^{n+1/2}$ according to the Boris scheme discussed above.

G. Wall Model

In a bounded discharge, majority of electrons is confined away from walls by the potential drop in the sheath. Only electrons with a sufficient energy to overcome the sheath drop will impact the wall. Electrons with insufficient energy to reach the wall will be reflected by the sheath. Due to surface roughness effects, the near wall sheath is expected to follow wall contours and be generally non-laminar. As such, it is possible that even electrons not actually reaching the wall will be diffusely scattered by the wall. This effect is not modeled in the present work and remains as a topic for future investigation. Instead, only particles actually impacting the wall are considered. In this work, the model of Sydorenko[13] was implemented. This model provides a curve fit to available experimental and analytic yields and takes into account elastic reflection, diffuse reflection, and yield of true secondary electrons. The secondary electron emission yield follows the analytical Vaughan fit.

The emission yields are plotted in Figure 4. This plot was generated by colliding 5,000 particles with a constant energy level W for each 1 eV increment in the 0 to 160 eV range. For $W < 13$ eV the wall yield is dominated by the elastic reflection of the primary electrons. Even though these reflection conserve energy, they can contribute to transport if the reflection is non-specular due to surface roughness. For $W > 30$ eV, the wall yield is dominated by the emission of secondary electrons, which exceeds unity at $W_1 = 52$ eV. High yield of secondary electrons can result in development of a non-planar saturated sheath which can trap emitted electrons in the near wall region. However, in the discharges studied in this work, $kT_e/e < W_1$.

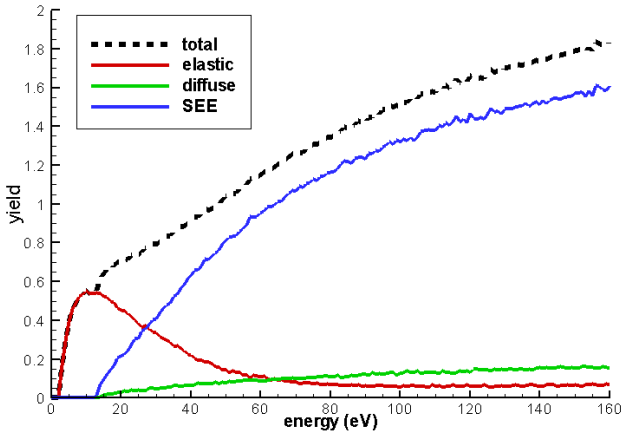


Figure 4. Variation in wall yield with incidence energy. This plot can be compared with Figure 3.4(a) in Sydorenko.[13]

Secondary electrons are injected into domain following the cosine distribution, $\vec{v} = v_M\hat{n}$ where

$$\hat{n}_0 = \sin(\theta)\cos(\xi) \tag{16}$$
$$\hat{n}_1 = \sin(\theta)\sin(\xi) \tag{17}$$
$$\hat{n}_2 = \cos(\theta) \tag{18}$$

with

$$\theta = \sin^{-1}(R_1) \tag{19}$$
$$\xi = 2\pi R_2 \tag{20}$$

and v_M is a velocity sampled from the Maxwellian distribution at the wall temperature. Maxwellian distribution was used instead of the perhaps more accurate effusion model for simplicity. R_1 and R_2 are two random numbers.

H. Particle Boundaries

Particles can leave the simulation domain in one of two ways: (a) by being lost to the walls or (b) by scattering sufficiently far from the magnetic field line. The first method is self-explanatory for a conducting wall. For the case of a dielectric wall, this loss corresponds to an electron attachment to the surface, contributing to the surface charge density, σ. The details of current collection are outlined in the following section.

The second method is merely a computational tool used to repopulated the initial velocity distribution function. In a real device, electrons are continuously traversing from one magnetic field line to another until they arrive at the anode. If the distance between field lines is taken to be r_L, the energy an electron gains in the transport is given by $W = E_\perp r_L$. For the Hall thruster discharge, this energy gain is on the order of one to several tens of eV and corresponds to the initial electron temperature at the field line. If an electron guiding center diffuses a significant distance from the starting position, it can be assumed to no longer correctly represent the population at the studied location. The size of the bounding envelope is somewhat arbitrary and can be set by the user. The size is set as $d = fr_{L,max}$ where $r_{L,max}$ is the Larmor radius computed at the smallest value of magnetic field. Although an argument could be made for using $f = 1$, such a value will immediately (assuming uniform B) remove all particles undergoing a scattering event. As such, f needs to be set to a sufficiently large value to allow secondary electrons to complete at least one oscillation between walls. Simulations presented in this work used $f = 5$.

I. Current Collection

Surface charge σ collected on the walls of a dielectric material contributes to the boundary conditions in the potential equation. The wall current at each wall consists of three terms: $j = j_i - j_e + j_s$ where the terms on the right hand side are the ion, primary electrons, and secondary electron current densities. In this formulation no distinction is made between true secondaries and reflected primary electrons. For a mirror-like wall, $j_e = j_s$ and the contributions cancel. The ion current density can be estimated from Bohm velocity. Note, this assumes that the sheath has not inverted; and that ions are entering the sheath in direction normal to the wall with the Bohm speed. This is often not the case, especially once novel thruster configurations utilizing magnetic lens are considered. SEE yield greater than unity can also result in an inverted sheath. Taking sheath inversion and sheath collapse into account remains as future work. In the simple Bohm model, the ion term becomes $j_i = en_0u_i$. For simplicity, $n_0 = \overline{n_i}$, the average ion density, is used. Again, determination of the actual sheath density remains as future work.

The electron terms are obtained from the kinetic electrons. In the code, the one dimensional volume scaling is used, $n = N/\Delta s$ in the bulk and $n = N/0.5\Delta s$ along the boundaries. $N = w_{sp}$ is the number of physical electrons carried by each electron macroparticle. Electron current density then becomes $j_e = 2ew_{sp}/\Delta s\vec{u} \cdot \hat{n}$. Similar expressions are written for the reflected and secondary populations. Surface charge collection begins at steady state and is incremented at each time step from $\sigma = \sigma + j\Delta$.

J. Collisions

Collisions are an important process affecting the diffusion of electrons. The collision types considered in the code are momentum transfer, ionization, excitation, and Coulomb collisions. The Monte Carlo Method (MCC) is used to determine collision probability. MCC is a fast method that is applicable in situations where the target population is not significantly affected by the collision events. MCC works by iterating through the list of source particles, and for each particle computing the collision probability. The probability is given by $P = 1 - \exp(-n_0\sigma_0 g\Delta t_c)$. Here σ_0 is the total collision cross-section, n_0 is the density of the target gas, g is the relative velocity (due to the high velocity of electrons, $g = v_e$), and Δt_c is the time difference between collisions. Typically, collisions are computed once every four time steps, $\Delta t_c = 4\Delta t$ for performance and statistical error reduction reasons.

The collision probability is compared to a random number R. If $P \geq R$, a collision occurs. Next, the code needs to determine which collision type occurred. This is done by calculating relative collision cross-sections, $\sum_{i=1}^{types} \sigma_i/\sigma_0 = 1$. The respective σ_i ranges are stored in an array such that $(\sigma_{i-1} : \sigma_{i-1} + \sigma_i)_i$.

A second random number is then drawn. Location of this random number in the array of relative sigmas is then determined, and the collision handler for the respective process is called. The collision handler operates solely on the particle velocities. Particle positions are not affected by the collision. Cross-sections for the considered collision types are plotted in Figure 5.

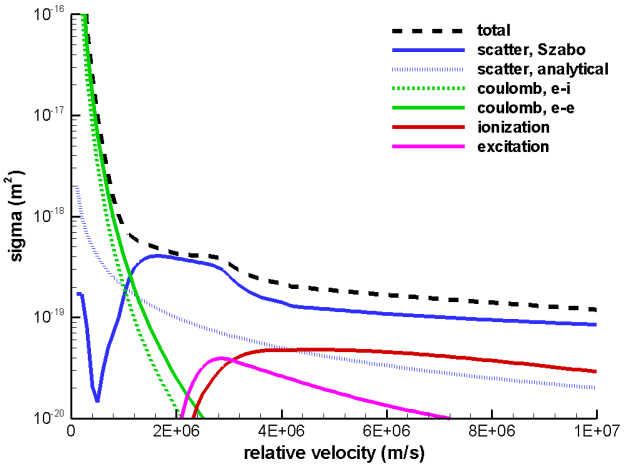


Figure 5. Collision cross-sections vs. energy. Scatter, excitation, and ionization models from Szabo.[19]

K. Momentum Transfer Collisions

An important collision process in Hall thruster discharges are the momentum transfer collisions between electron and neutral atoms. These collisions are due to electrostatic interaction between the electron and the atom, where the interaction forces arises from the polarization of the atom. An analytical model for polarization scattering is given by Lieberman,[20]

$$\sigma_L = \pi b_L^2 = \left(\frac{\pi \alpha_p q^2}{\epsilon_0 m_R}\right)^{1/2} \frac{1}{g} \tag{21}$$

where $\alpha_p = \alpha_R a_0^3$ is the polarizability (in units of volume), α_R is a proportionality constant, and $a_0 = 53$ pm is the Bohr radius. The reduced mass m_R is given by $m_R = m_e m_a/(m_e + m_a) \approx m_e$ for electron-atom collisions. The relative velocity $g = |v_e - v_a| \approx |v_e|$. The relative proportionality is obtained using available Xenon polarization data, which list $\alpha_p = 4$ Å^3, resulting in $\alpha_R = 26.9$. However, when this model is compared to experimental measurements, it is seen to underpredict the collision cross-section by approximately half order of magnitude except in the low energy region, where it dominates over the experimental fit. The fit to experimental data, as utilized by Szabo[19] is seen in Figure 5 in blue. The analytical polarization profile is shown with the dotted blue curve. For this reason, Szabo's model is used in this work to compute the cross-section. The expression for the curve is available in the Appendix in.[19]

Electron scatter collisions are modeled by colliding the electron with a virtual target neutral sampled from the background population at the background temperature. The algorithm is based on the approach described by Bird.[21] The algorithm first computes the center of mass and a random deflection angle. The post-collision electron velocity is then updated from

$$\vec{v}^* = \vec{v}_m + \frac{m_2}{m_1 + m_2} g^* \tag{22}$$

where $\vec{v}_m$ is the center of mass velocity and g^* is the post collision relative velocity vector. It should be noted that the magnitude of g is preserved in the collision, $|g| = |g^*|$.

L. Ionization and Excitation Collisions

Szabo's models are also used for the ionization and excitation collisions. This model uses polynomial fits to experimental data which are provided in the Appendix of 19. These models are not included here for

brevity. Ionization collisions occur only for electrons with energy grater then $W_{iz} = 12.1$ eV. The energy of the colliding electrons is reduced by W_{iz} and their post collision direction is assumed to diffuse in the 3D space. Similarly, for excitation collision, $\sigma = 0$ for $W_{ex} < 8.12$. The collision process is modeled by reducing the electron energy by a random amount for $W_{ex} < R < W_{iz}$. Again, the post collision direction is selected randomly.

M. Coulomb Collisions

As indicated by Figure 5, Coulomb collisions are an important process at low energies. Unlike the previously considered collisions, Coulomb collisions arise from the electrostatic attraction or repulsion between charged particles and thus the interaction force magnitude scales with the inverse of the distance. Without considering an upper bound, these collisions would result in an infinitely large cross-section. For this reason, λ_D is selected as the typical upper bound over which the interaction force is integrated. This then leads to a very large cross-section, $\sigma \sim \pi\lambda_D^2$. However, vast majority of Coulomb collisions are small angle events. A common approach is to approximate the large number of small angle collisions with a corresponding large angle cross-section. Lieberman[20] gives the large-angle cross-section as

$$\sigma_{90} = \frac{8}{\pi} b_0^2 \ln \Lambda \tag{23}$$

where $\ln \Lambda \approx 10$ for typical plasma discharges and b_0 is the distance of closes approach, given by

$$b_0 = \frac{q_1}{q_2} 4\pi\epsilon_0 W_R \tag{24}$$

where $W_R = \frac{1}{2} m_R v_R^2$ is the center of mass kinetic energy.

Since this cross section approaches infinity as $v_r \to 0$, the code limits minimum v_r to 10^4 m/s. Electron-ion Coulomb collisions are modeled using the same algorithm utilized for electron-neutral momentum transfer collisions, where a virtual target ion is sampled from the background population. This approach is not valid for electron-electron interactions. Electron-electron Coulomb interactions are the primary mean by which electron velocity distribution functions thermalize and hence it is important to consider both source and target particles in the collision event. A modified MCC algorithm was implemented to handle this interaction event. Prior to computing collision probabilities, electrons are grouped by cell. Next, during a collision event, a target electron is picked randomly from the electrons located in the same cell as the source, assuring the target particle is different from the source. Collision is then performed with these two particles, and velocities of source and target are updated. Since this effectively results in a doubling of collisional rate due to the MCC algorithm, the collision cross-section for electron-electron collisions is reduced by half, $b_{0,eff} = 0.5 b_0$.

N. Transport Calculation

The primary output from a Lynx simulation is the spatial variation in mobility along the magnetic field line, $\mu = \mu(s)$. Conceptually, the average drift velocity can be computed by averaging the particle perpendicular component of velocity. Mobility can then be determined from $\mu = v_d/E_\perp$. However, such an approach does not work in practice due to statistical errors. The simulation noise in the computation of the average drift velocity is on the order of μ. This noise arises due to the discretized nature of the particle push, the particle traverse along the field line, and the averaging of velocities at nodal positions.

Instead, the drift velocity is computed from the speed with which guiding centers diffuse across the field line. The x position of the guiding center, the point about which the particle orbits in the $x - y$ plane, center can be computed from $x_{gc} = 0.5(x_{min} + x_{max})$, where the two positions on the right hand side are the minimum and maximum x position attained by the particle. These values are continuously updated by the particle integrator. Scattering events, such as collisions or wall interactions, result in the particle orbit shifting to a new guiding center, which will be demonstrated by a shift in the x range. Guiding center positions are checked once per orbit, and only particles that have completed at least one orbit since birth are considered. Particle x_{min} and x_{max} values are reset during this operation. Particles undergoing a transport event in the previous cyclotron orbit will have a guiding center intermediate of the initial and the final position. The correct shift will be computed upon the subsequent calculation (assuming no additional scattering events) due to particle x range reset. Once the position of the guiding center is known, the drift

speed is obtained from $v_{gc} = \Delta x_{gc}/(t_{now} - t_{born})$. The term on the right hand side is the time the particle has been alive. Since in the code the initial $x_{gc} = 0$, the shift is given by the actual guiding center position. Mobility is then computed from $\mu = v_{gc}/E_{\perp}$.

III. Starfish

Our previous effort[12] utilized HPHall as the TPM solver. While HPHall allowed us to perform a proof-of-concept study, a number of issues made it unsuitable for future use. First, HPHall is written in the structured C programming language. Modern programming practices follow the object-oriented design as it leads to an increased modularity and ease of maintenance. Our solvers are developed in the strongly objected-oriented Java programming language. While coupling of C and Java is feasible, it is both non-trivial and impractical. Secondly, the lack of a source code repository resulted in a number of competing versions of HPHall. In addition, new models were added via compiler directives. Without a rigorous analysis of the source code and the set of compiler flags used to build the exucatble, it is impossible to correlate simulation results to a particular set of documented equations.

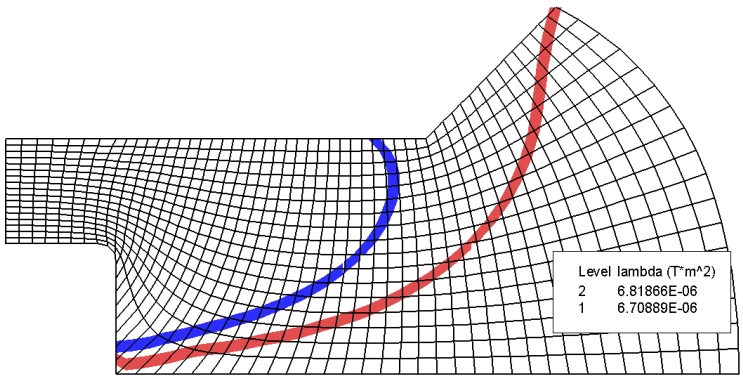


Figure 6. The desired (red, right) and the used (blue, left) cathode λ line.

We also encountered difficulties in applying HPHall to a non-annular thruster geometry. This issue is illustrated in Figure 6. The magnetic field line on the right, shown in red, corresponds to the cathode boundary as specified in the input file. For reasons not fully understood, HPHall adjusts the specified cathode λ such that the field line intersects the inner wall at a grid node. Due to the convergence of magnetic field lines at the centerpole of the CHT, a small offset in λ results in a significant shift of the actual boundary. The boundary, as used by the code, is shown by the blue line. Our effort to expand the boundary into the downstream direction was not successful. Since HPHall solves electron equations only up to the cathode boundary, the inability to capture the cylindrical region excluded a significant portion of the thruster from the detailed analysis. We have thus started development of a new 2D solver called Starfish.[22] Although in the present analysis Starfish serves the role previously occupied by HPHall, Starfish is being developed as a general 2D plasma / rarefied gas solver with applications extending past Hall thruster analysis and including processes such as plume modeling, plasma processing, contamination transport, and atmospheric discharges.

A. Code Structure

In development of Starfish we utilized lessons learned from the AFRL Coliseum project.[23] The strength of Coliseum lies in its flexibility, and we were interested in capturing this feature in Starfish. Starfish is based on the concept of a "runnable module". This module is a Java class that can be associated with a specific X`<element>` in the XML input file. Each concrete implementation of the RunnableModule base class implements a "process" function which is called whenever an XML element with the associated name is encountered in the input file. The module then parses the user input and constructs appropriate data structures. The object oriented design allows the main loop to manipulate these data structures without requiring knowledge of their implementation.

The typical Starfish simulation input file is shown below. First several external files are loaded. The

parsing of the input file is performed in two steps, with the first step simply replacing each `<load>` element with the content of the file. This allows for an arbitrary separation of input settings, while also simplifying reuse of common datasets such as material definitions or interactions lists. The content of these files is discussed below. Potential solver parameters are specified next. This particular example utilizes the thermalized electron model which is discussed in more detail below. Additional solvers implemented so far include a linear and a non-linear Poisson solver, as well as the quasineutral Boltzmann inversion model. The coefficient matrix for the Gauss-Seidel solver is obtained with the finite volume approach, allowing the code to utilize non-rectilinear grids. The input file next specifies time control parameters. These include the number of time steps along with the time step size, in seconds. The simulation is then executed with the `<starfish />` command. Finally, simulation results are outputted in the Tecplot format. The code contains handlers for both 2D and 1D outputs. The 2D output is used for save results on the computational grid, while the 1D output saves results along the boundary splines.

```
<simulation>
<!-- load input files -->
<load>materials.xml</load>
<load>boundaries.xml</load>
<load>domain.xml</load>
<load>sources.xml</load>
<load>interactions.xml</load>

<!-- load external magnetic field data -->
<load_field format="tecplot" name="bfield">
<file_name>2d_ave_tp.dat</file_name>
<coords>z,r</coords>
<vars>bfi=bz,bfj=br,lambda</vars>
</load_field>

<!-- set potential solver -->
<solver type="thermalized">
<bottom>symmetry_outside,symmetry_inside,inner_wall</bottom>
<top>outer_wall,thruster</top>
<anode_lambda>0</anode_lambda>
<cathode_lambda>6.79998e-06</cathode_lambda>
<nodes>10,12</nodes>
<cathode_kTe>5</cathode_kTe>
<cathode_phi>100</cathode_phi>
<power>100</power>
</solver>

<!-- set time parameters -->
<time num_it="0" dt="5e-8" />

<!-- run simulation -->
<starfish />

<!-- save results -->
<output type="1D" file_name="flux.dat" format="tecplot" />
<variables>influx.xe+, outflux.bn, erosion_rate</variables>
</output>

<output type="2D" file_name="field.dat" format="tecplot">
<variables>phi, t.e-, nd.xe, nd.xe+, u.xe+, v.xe+</variables>
</output>
</simulation>
```

B. Domain

The domain element is used to specify the extent of the computational domain. Starfish supports multiple meshes. The present version leaves the responsibility of assuring the meshes describe a contiguous domain with the user. Starfish supports only topologically structured meshes, however, the meshes can be either rectilinear or quadrilateral. A quadrilateral mesh is formed by specifying a list of boundary splines forming each of the four boundaries, as well as the number of nodes in the two spatial directions. The node positions are then calculated in the internal domain by solving the elliptic equations $\nabla^2 l = 0$ and $\nabla^2 m = 0$ where l and m are the two logical coordinates. The physical to logical mapping on the local cell-level of a quadrilateral

mesh is

$$x = \alpha_1 + \alpha_2 l + \alpha_3 m + \alpha_4 lm \tag{25}$$
$$y = \beta_1 + \beta_2 l + \beta_3 m + \beta_4 lm \tag{26}$$
$$\tag{27}$$

with the coefficients given by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \tag{28}$$

The x terms on the LHS correspond to the cell vertices.

As an example, the code snippet below specifies a uniform rectilinear and a quadrilateral elliptic mesh. It should be noted, that in this paper we used a computational domain consisting solely of rectilinear meshes.

```
<domain type="rz">

<mesh type="uniform" name="annular">
<origin>0.0491, 0.0071</origin>
<spacing>5e-4, 3.35e-4</spacing>
<nodes>13, 18</nodes>
</mesh>

<mesh type="elliptic" name="downstream">
<left>exit_plane</left>
<bottom>symmetry_outside</bottom>
<right>downstream</right>
<top>thruster</top>
<nodes>35,13</nodes>
</mesh>
</domain>
```

C. Boundaries

Surface boundaries are specified in the `<boundaries>` element. Each surface in Starfish is described by a linear or a cubic Bezier spline. The spline path uses syntax similar to the SVG (Scalable Vector Graphics) notation, with the inclusion of special connector tags. Each spline must be given a name as well as material if the spline describes a solid surface. The example below shows a linear and a cubic spline. A linear spline consists of nodes joined by straight lines. The "M" (move), "L" (linear), and "C" (cubic) commands simply act to change the current mode, in a fashion analogous to SVG, and thus it is not necessary to repeat the "L" command between segments. The `inner_wall` spline consists of two segments connecting three nodes.

```
<boundaries>
<boundary name="INNER_WALL" type="solid">
<material>BN</material>
<path>M 0.0551,0 L 0.0551,0.0071 anode1:first</path>
</boundary>

<boundary name="DOWNSTREAM" type="open">
<path>M thruster:last C 0.0853587,0.0146086 0.0884662,0.0069188 0.0891,0</path>
</boundary>
</boundaries>
```

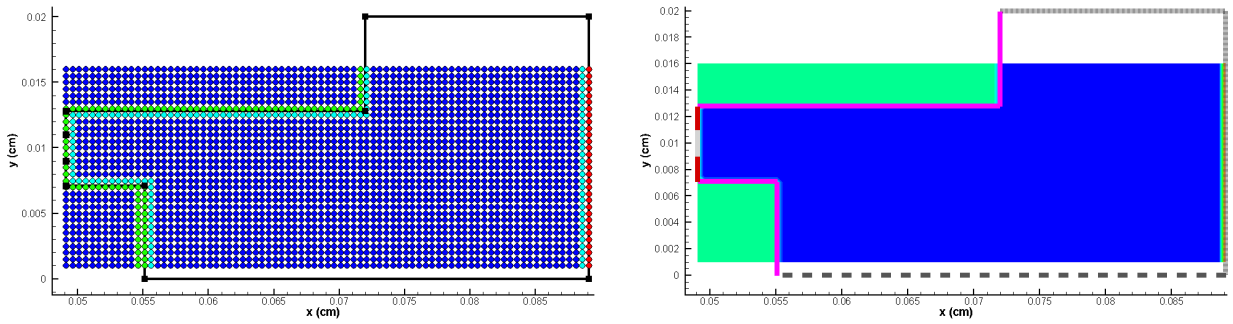
The syntax for Bezier splines differs from the SVG implementation. While in SVG, cubic splines are specified by providing the two end points and the two knots for the spline, in Starfish, the cubic spline only specifies the points through which the spline passes. The code automatically determines the knot locations by solving the following system,

$$\mathbf{B}_i(0) = \mathbf{B}_{i-1}(1) \tag{29}$$
$$\mathbf{B}'_i(0) = \mathbf{B}'_{i-1}(1) \tag{30}$$
$$\mathbf{B}''_i(0) = \mathbf{B}''_{i-1}(1) \tag{31}$$
$$\tag{32}$$

where

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2 t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, \quad t \in [0,1] \tag{33}$$

is the cubic Bezier spline.[24] The boundaries serve several important roles. They are used to define particle sources and to specify surface impact interactions. The boundaries are also used to specify boundaries for the potential solver. Starfish contains an internal meshing module for intersecting boundaries with the mesh and classifying nodes as solid or fluid. Starfish uses a topologically structured mesh, since such a mesh simplifies the particle push operation. In a structured mesh, an analytical expression exists between the particle physical and logical locations, allowing the code to determine the particle cell location through a mapping function. The downside of structured approaches is that cell boundaries may not line up with the physical domain. As such, Starfish uses cut cells to determine surface boundary. Each NCV includes the surrounding cell in each mesh direction. Next, the code determines which spline segments reside in each NCV. These segments are subsequently used to classify the node location as internal (solid) or external (fluid) to the node. The location is based on the direction of node ordering. The initial node classification for the CHT geometry is shown below in Figures 12 and 13. Once the initial nodes are classified, Starfish performs a flood-fill operation to classify the remaining nodes.



(a) Interface node classification

(b) Flood fill

Figure 7. Illustration of the Starfish node classification approach

D. Materials

The listing below shows the material file which is used to define simulation materials. Materials define the properties of all components and chemical species present in the simulation. As this example shows, multiple material types are available.

```
<material>
<material name="Xe+" type="kinetic">
        <molwt>131.3</molwt>
        <charge>1</charge>
        <spwt>5e9</spwt>
        <init>nd=1e18,nd_back=1e4</init>
</material>

<material name="SS" type="solid">
        <molwt>52.3</molwt>
        <density>8000</density>
</material>
```

Materials are an example of the benefit of object oriented design. Each concrete material implementation implements an `updateFields` function. The objective of this function is to compute the density, temperature, and velocity of the material at the current time step. The actual implementation of this function is material-type specific. Among the implemented material types are solid, kinetic,and fluid materials. Solid materials are used to define the domain occupied by the the physical object and thus their density does no chage. The other types form flying materials, gas-like materials that can change their density. The kinetic material defines its population by simulation macroparticles, with `updateFields()` performing the standard particle-in-cell (PIC) push. A fluid material on the other hand updates density by integrating fluid equations

through the simulation time step. Currently only a neutral Navier Stokes algorithm is implemented; a magnetohydrodynamic solver for ionized gases will be added in the near future.

E. Sources

The code listing below shows the XML file used to specify sources. Sources are basically initial conditions in the velocity space. Starfish contains four types of particle sources: surface boundary sources, surface material source, volume sources, and surface particle list sources. Currently only the first two kinds, the surface boundary and surface material source, can be specified by the user. The other two sources are used internally in the code to perform surface interactions or to create particles due to chemical reactions. Sources can sample either particles or fluid (by modifying the boundary values). A surface source is attached to a particular named boundary. The source is given a certain velocity emission model, such as uniform or drifting Maxwellian. The emission material is also specified. The sources inject particles along the normal direction of each segment making up the associated spline. Particles are loaded at a random location along the spline. The actual source parameters, such as temperature, or drift speed, are emission model specific. A material source is similar to a boundary source with the exception that it attaches itself to all boundaries composed of the specified source material. Such a source can be used to model outgassing. Currently only a time-independent mass flow rates are available, however, support for time-dependent sourceswill be added in the future.

```
<sources>
<boundary_source name="neutral_source" type="maxwellian">
<material>xe</material>
<boundary>inlet</boundary>
<mdot>4e-6</mdot>
<v_drift>300</v_drift>
<temp>1000</temp>
</boundary_source>
</sources>
```

F. Material Interactions

The final file lists material interactions. The input file is shown below. Four types of material interactions are available in Starfish: chemical reaction, MCC collision, DSMC collision, and surface impact. The first is an interaction between two density fields. This model is applicable to both fluid and kinetic materials, since the kinetic particles are merely used to update the density map of the material. As shown below, chemical reactions are specified by listing the source and product materials along with optional coefficients. The "model" corresponds to the rate equations. For the ionization reaction, $Xe + e- \rightarrow Xe^+ + 2e-$, Fife provides the rate equation

$$\dot{n}_i = \chi(T_e) n_e n_a \tag{34}$$

where $\chi(T_e)$ is obtained by integrating a differential cross-section.

The MCC collision is an interaction between a particle and a density field. The source material for this interaction type must be kinetic, while the target may be fluid or kinetic. MCC is typically used to model collisions with a significantly denser target cloud which can be assumed to be only negligible affected by the collisions. An example of this interaction in the HET is the charge exchange (CEX) reaction between ions and neutrals. The MCC model is not suitable for interacting two like populations, since it does not modify the target particle. DSMC collision is an interaction between two kinetic materials, and is suitable for modeling ion-ion collisions. Finally, surface impact is an interaction between a flying and a solid material. This interaction type specifies the post-impact material, as well as sticking, restitution, and accommodation coefficients. A sputter model can also be specified and can be coupled with a real-time surface deformation.

```
<material_interactions>
<chemical_reaction model="ionization">
        <sources>Xe,e-</sources>
        <products>Xe+,2*e-</products>
</chemical_reaction>

<mcc_collision process="scatter">
        <source>Xe+</source>
        <target>Xe</target>
```

```
        <sigma>const</sigma>
        <sigma_coeffs>3e-19</sigma_coeffs>
</mcc_collision>

<surface_impact source="Xe+" target="BN">
        <emission>diffuse</emission>
        <product>Xe</product>
        <c_stick>0.5</c_stick>
        <c_rest>1</c_rest>
        <c_accom>0.5</c_accom>
        <sputter type="const" yield="0.1" product="BN" />
</surface_impact>
</material_interactions>
```

G. Thermalized Potential Solver

Starfish contains several field solvers, including a Poisson solver and the thermalized potential solver. The latter directly mirrors the implementation of Fife 25. First, in the direction along the magnetic line, the temperature is assumed to remain constant. Since plasma is assumed isothermal, any magnetic mirror drops out,[26] and the force balance may be written,

$$\nabla_{\parallel} P_e = \nabla_{\parallel}(n_e k T_e) = e n_e \nabla_{\parallel} \phi \tag{35}$$

which is integrated

$$\phi - \frac{kT_e}{e}\ln(n_e) = \phi_0 - \frac{kT_e}{e}\ln(n_0) \tag{36}$$

In HPHall, the following expression is used instead, alleviating the need to define a reference density

$$\phi - \frac{kT_e}{e}\ln(n_e) = \phi^*(\lambda) \tag{37}$$

The ϕ^* term is the thermalized potential. Knowing ϕ^* and kT_e along each field line allows us to determine the spatial variation in potential, ϕ from ion density n_i. Quasineutrality is assumed, $Zn_i \approx n_e \approx n$, which is a valid assumption in the bulk region removed from the near-wall sheath. This approach reduces the dimensionality of the electron solve from the 2D $r-z$ domain to a 1D λ domain, where λ is the magnetic field line.

The other term that is necessary in order to determine potential is the electron temperature. In fact, the electron fluid model implemented in HPHall can be primarily thought of as a temperature solver. The electron temperature is given by the energy equation,

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_e k T_e\right) + \nabla\cdot\left(\frac{5}{2}n_e k T_e \vec{u}_e + \vec{q}_e\right) - \vec{u}_e\cdot(n_e k T_e) = S_e - S_i \tag{38}$$

where S_e and S_i are energy losses due elastic and inelastic collisions with neutrals and ions, and the heat term $\vec{q}_e$ also includes energy losses to the wall. These properties are area averaged over the magnetic field line.

The initial electron cross-field velocity is obtained from the force balance presented in Equation 35

$$j_{e,\hat{n}} = -e n_e u_{e,\hat{n}} = \sigma_{e,\perp}\left(E_{\hat{n}} + \frac{1}{e n_e}\frac{\partial p_e}{\partial \hat{n}}\right) \tag{39}$$

or

$$u_{e,\hat{n}} = -\mu_{e,\perp}\left(E_{\hat{n}} + \frac{1}{e n_e}\frac{\partial p_e}{\partial \hat{n}}\right) \tag{40}$$

This expression is further simplified using $p = n_e k T_e$ and $\vec{E} = -\nabla\phi$ to obtain

$$u_{e,\hat{n}} = \mu_{e,\perp}\left(\frac{\partial \phi}{\partial \hat{n}} - \frac{kT_e}{e n_e}\frac{\partial n_e}{\partial \hat{n}} - \frac{k}{e}\frac{\partial T_e}{\partial \hat{n}}\right) \tag{41}$$

Fife further simplified this expression by differentiating 37 to obtain an expression in terms of the thermalized potential

$$u_{e,\hat{n}} = \mu_{e,\perp}\left(\frac{\partial \phi^*}{\partial \hat{n}} + \frac{k}{e}\left(\ln(n_e) - 1\right)\frac{\partial T_e}{\partial \hat{n}}\right) \tag{42}$$

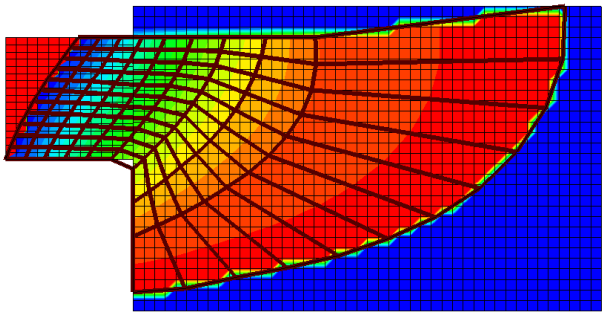


Figure 8. Illustration of the electron lambda mesh overlaid over the ion mesh

Here $\mu_{e,\perp}$ is the mobility term which in HPHall is evaluated following a model analogous to Equation 2.

At present, Starfish does not contain a magnetic solver. Instead the code reads the results Tecplot file from HPHall which contains the magnetic field and λ values stored on the HPHall quadrilateral mesh. While Starfish supports a similar quadrilateral mesh, in the work presented here we utilized a Cartesian mesh for the ions. The loaded magnetic field is first interpolated onto this mesh. The code next constructs the lambda mesh by countouring the λ field. Contouring begins along the spline listed by the ¡start¿ element and continues until the ¡end¿ spline. The marching squares algorithm is used to perform the contouring. The contour is subsequently desampled to the user specified number of radial partitions. Unlike HPHall, the Starfish lambda domain is a topologically structured grid with a fixed number of radial partitions. Figure 8 shows an example of the constructed lambda mesh. The values from the lambda mesh are also interpolated onto the ion mesh. Constant potential is applied upstream of the anode boundary. The linear decay downstream of the anode is not included in this plot.

H. Integration with Lynx

Since both Starfish and Lynx are developed in the Java programming language, the two codes can be integrated easily by include the Lynx .jar file in the Starfish project. Lynx can be launched as a standalone program or as a module. In the second case, the input files are passed to the launcher as function arguments instead of being loaded from the file.

IV. The Wall-Plasma Interface

The next aspect of the multiscale approach deals with the near wall region, especially as applicable to novel thruster geometries. In the classical Hall thruster, the magnetic field consists primarily of the radial component. Such a configuration appears ideal at first since it produces an electric field directed along the thruster axis. However, the presence of walls modifies the near-wall potential structure and results in a local component accelerating ions into the walls. Ion wall flux contributes to a loss of thruster efficiency and to a limited thruster lifetime due to channel erosion. In order to mitigate these losses, some novel Hall thrusters[9,27] have begun experimenting with magnetic fields with convex geometry. Near the walls, this so-called magnetic lens induces an electric field with a radial component directed towards the channel centerline.[28] An interesting aspect of the lens configuration is that in the vicinity of the wall, the resulting magnetic field lines can approach the wall with a highly inclined incidence angle θ, as measured from the wall normal. Such a configuration leads to an electric field with a strong radial term that, in the case of a sufficiently large θ, dominates the component due to the sheath potential drop.[29] This can be seen from a simple example. Consider a typical 300V Hall thruster with a 200V potential drop occurring across a 1 cm wide acceleration zone. The magnitude of the electric field $E_\perp$ is then 2×10^4 V/m. Next consider the potential drop due to the wall sheath. The electric field along the magnetic field line in the vicinity of the wall can be estimated from $E_\parallel = T_e \partial \ln n / \partial r \sim T_e / \Delta r \sim 20$ eV / 0.1cm $\sim 2 \times 10^5$ V/m.[30] Here Δr is the sheath thickness, which is taken to be 10 Debye lengths. The angle at which the radial component of the electric field becomes negative is given by $E_r = E_\parallel \cos\theta - E_\perp \sin\theta \sim \tan\theta = E_\parallel / E_\perp$ or $\theta \sim 85°$.

Ions are then accelerated away from the wall and a complete sheath collapse is expected. The thermalized

potential model does not contain sufficient resolution to resolve the near-wall non-neutral region. We have thus developed a code specializing in modeling ion motion in the sheath region. This code allows us to investigate the sheath formation and collapse in the presence of a two dimensional magnetic field. This analysis is performed using a simple axisymmetric electrostatic particle-in-cell (ES-PIC) code.

A. Numerical Model

The code is based on the hybrid approach in which ions are treated as particles, but electrons are represented by a fluid model. The computational domain is limited to a small region near the outer wall, as illustrated in Figure 9. The small size of the computational domain allows the code to resolve the Debye length and thus directly compute the electric potential in a reasonable amount of time (each simulation takes approximately 30 minutes). The domain captures the acceleration region characterized by the presence of strong applied magnetic field. In this formulation, the anode and the primary ionization zone are located to the left. The upper boundary represents the wall, while the bottom boundary extends into the quasineutral bulk plasma region. Ions are injected into the simulation along the left boundary and leave through the open right and bottom face or by recombining with the upper wall. The inset in Figure 9 is an example of a hybrid annular/cylindrical Hall thruster such as the CHT in which the magnetic field geometries of interest can be found. The highlighted box illustrates the location of the simulation domain. It should be noted that the size of the region of interest is artificially increased in this drawing for clarity.

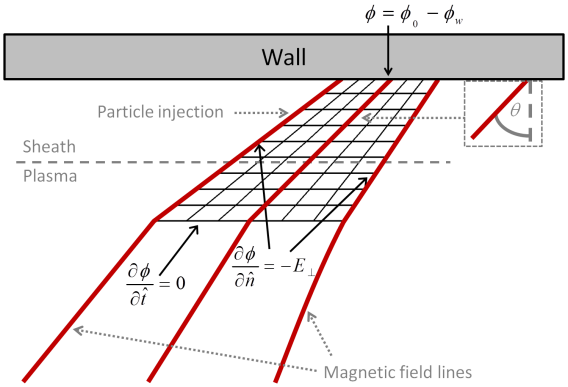


Figure 9. Schematic of the computational domain. Ion particles are injected from the left. The inset shows a cylindrical Hall thruster and highlights the region analyzed by our code.

To simplify the subsequent computation, a simulation mesh is selected such that radial gridlines are aligned with the magnetic field. Such a formulation allows us to specify the necessary reference values as a function of the axial grid coordinate only. At this time, the computational mesh is described analytically instead of utilizing the mesh from the thruster simulaton. The mesh coordinates are given by

$$r = r_0 + j * \Delta r \tag{43}$$

$$z = i * \Delta z_j - (nr - 1 - j) * \Delta r * \tan(\theta) - 0.5 * (nz - 1) * (\Delta nz_j - \Delta z_w) \tag{44}$$

where Δz_j is the local cell spacing. The cell spacing varies linearly between the top and bottom boundary. These mesh coordinates can be easily inverted. The j component is obtained first from the radial r coordinate. The i coordinate is then recovered from the axial position z using the second equation.

Xenon ions are injected into the simulation domain along the left boundary with initial velocity $u_z = u_0 + u_{th}$. Here u_0 is the drift component and u_{th} is a random thermal velocity obtained by sampling the Maxwellian distribution function at 1 eV. The magnitude of the drift component was set to 6 km/s, corresponding to approximately 25 eV of upstream acceleration. Initial radial velocity is also obtained by sampling the random thermal component. The number of computational particles injected per time step is obtained from $p = \dot{m}\Delta t/w = n_i \bar{u} A m \Delta t/w$ where $n_i = 5 \times 10^{16}$ m^{-3} is the injection ion density, and w is the macroparticle weight. The weight was selected such that cells in the bulk region contained approximately 200 computational particles at steady state. Particles were loaded with a zero azimuthal component. We assume that no forces act in the azimuthal direction and hence the cylindrical equations of motion reduce

to the Cartesian form. Ion positions are updated at each time step according to the Leapfrog algorithm by integrating the Lorentz force, $\vec{F} = -e\nabla\phi$. The magnetic term is omitted, since in a Hall thruster, ions are not magnetized. Ions impacting the upper wall or leaving the computational domain were removed from the simulation. Collisions were not included as they generally play only a minor role in the sheath.

B. Potential Solver

Potential was computed by solving the Poisson's equation, $\epsilon_0\nabla^2\phi = -e(n_i - n_e - n_s)$, with the three densities on the right side corresponding to ions, primary electrons, and secondary electrons, respectively. The ion density n_i was obtained by scattering positions of kinetic ions to the computational grid. The electron density is computed from the Boltzmann equation following the approach in[26] and.[31] In the frame of reference of ions, electrons respond instantaneously to a disturbance. The time-dependent and convective terms then vanish from the momentum equation. Also, since in Hall thrusters the sheaths is generally collisionless, we can disregard the collision operator. We thus arrive at the force balance,

$$\frac{\partial}{\partial x}(n_e k T_{e\|}) + \frac{n_e(kT_{e,\perp} - kT_{e\|})}{B}\frac{\partial B}{\partial s} + n_e e E_{\|} = 0 \tag{45}$$

These terms correspond to the gas pressure, magnetic mirror, and electric field effects, respectively. Utilizing $E_{\|} = -\partial\phi/\partial s$, the above equation can be integrated to obtain an expression for bulk electron density,

$$n_e = n_0 \exp\left[\frac{e}{kT_{e\|}}(\phi - \phi_0) - \frac{kT_{e\perp} - kT_{e\|}}{kT_{e\|}}\ln\left(\frac{B}{B_0}\right)\right] \tag{46}$$

This is the well known Boltzmann relationship modified by the magnetic field strength term. This term is seen to reduce the electron density in regions of an increasing magnetic field this is the magnetic mirror effect. The standard Boltzmann relationship is recovered if the magnetic field magnitude remains constant along the field lines. The magnetic mirror term also drops out if plasma is isothermal. However, as outlined in,[7] Hall thruster plasma is not isothermal. Following the results from the kinetic analysis in,[7] $kT_{\|} = 10$ eV and $kT_{\perp} = 2kT_{e\|} = 20$ eV is used. For this particular set of input parameters, Equation 46 simplifies to $n_e = n_0(B_0/B)\exp[e(\phi - \phi_0)/kT_{e\|}]$.

It should be noted that Equation 46 holds independently for each magnetic field line. The three constants with the 0 subscript are the reference density, potential, and magnetic field strength. These values are unique and independent along each line. We also assume that electron temperature remains constant in the parallel direction, $\partial T_e/\partial s = 0$, and that there is no variation in magnetic field strength in the axial direction, $\partial B/\partial z = 0$. The reference density is obtained self-consistently from the computed ion density along the bottom edge of the simulation domain where $n_i = n_e = n_0$. A linear decay in potential is applied for majority of cases, with $\phi_0 = \phi_L - E_{\perp}(z_w - z_{w,0})$, where $E_{\perp} = 20$ kV/m. Since in Hall thrusters the potential profile adjusts self-consistently based on the local discharge parameters, an alternate cusp configuration is also investigated in which the potential profile exhibits a deep valley. The strength of the magnetic field is computed from the conservation of magnetic flux, $\phi_m = \int_S \vec{B}\cdot\vec{ds}$ or $Br\Delta z = C$, a constant value. Here Δz is the cell spacing at the corresponding r value. As indicated by Equation 46, terms relating to the magnetic strength appear only as a ratio allowing us to select an arbitrary value for the reference field.

The secondary electron density n_s is obtained from $\nabla\cdot(n\vec{u}) = 0$. Density of secondary electrons at the wall is given by $n_{s,w} = sn_{e,w}$ where $s(T_e,\theta)$ is the SEE yield.[30] The electrons are assumed to be emitted with an isotropic angular distribution and energy dependence based on the linear relationship given by Dunaevsky,[32]

$$s(T_e,\theta) \approx \sigma_0 + (1-\sigma_0)\frac{E_p}{E_1} \tag{47}$$

For Boron Nitride, the typical wall material in conventional Hall thrusters, the coefficients σ_0 and E_1 are 0.54 and 40, respectively. E_p is the energy of the incoming particle, measured in eV. Initial velocity of the secondary electrons is taken to be $u_{s,w} = (2kT_w/\pi m_e)^{1/2}$. Energy conservation dictates $u = (2q\Delta\phi/m)^{1/2}$, leading to

$$n_s = sn_{e,w}\left(\frac{kT_w}{e\pi}\frac{1}{\phi - \phi_w}\right) \tag{48}$$

Potential along the top wall is fixed as $\phi_w = \phi_0 - \Delta\phi_w$, where the wall potential drop is given by[30] as

$$\Delta\phi_w = T_e \ln\left\{\frac{1 - s(T_e, \theta)}{v_0\left(\frac{2\pi m}{T_e}\right)}\right\} \tag{49}$$

where v_0 is the ion velocity at the sheath edge, which in this formulation is set to the Bohm speed. The problem is closed by prescribing the normal electric field $E_\perp$ along the left and right boundaries, and zero tangential electric field $E_\perp = 0$ on the bottom boundary. The electric field along the left and right boundaries is non-uniform for cases with a diverging magnetic field line topology. This can be seen from a simple observation of the increasing distance between field lines as one moves away from the wall. The magnitude is obtained numerically by computing the normal distance d to the next magnetic field line (grid line) at each node. The electric field is then set from $E_\perp = -\Delta\phi_0/d$. Potential is solved using the finite volume method.

In addition to the Poisson solver, an alternate method for obtaining potential was also implemented. This approach was developed in order to approximate the solution from Hall thruster codes such as HPHall. As described previously, HPHall does not solve the Poisson's equation, and instead computes radial potential by assuming quasineutrality, $n_e = n_i$ (with ion density obtained from kinetic ions) in conjunction with the thermalized potential model, $\phi = \phi^* + kT_e/e \ln(n/n_0)$. This approach is analogous to the formulation used to derive the relationship for bulk electron density, Equation 46. This expression can be inverted to obtain

$$\phi = \phi_0 + \frac{kT_{e\parallel}}{e} \ln\left(\frac{n}{n_0}\right) + \frac{kT_{e\perp} - kT_{e\parallel}}{e} \ln\left(\frac{B}{B_0}\right) \tag{50}$$

where $n_e = n_i = n$. Our expression extends the quasineutral formulation by taking into account the magnetic mirror term.

C. Near Wall Sheath Formation

The plasma-wall transition region in the Hall thruster channel determines the particle and energy fluxes from the plasma to the wall. In order to develop a self-consistent model, the boundary parameters at the sheath edge (ion velocity and electric field) have to be obtained from a multi-dimensional (in our case, two-dimensional) plasma bulk particle model. Under typical Hall thruster conditions, i.e. ion temperature much smaller than that of electrons and significant ion acceleration in the axial direction, the pre-sheath scale length becomes comparable to the channel width so that the plasma channel becomes an effective pre-sheath. We have shown previously that the plasma-sheath patching approach can be used. In this approach, the electric field that develops in the pre-sheath can serve as a boundary condition for the sheath in addition to the ion velocity. Relationship between the ion velocity and electric field at the plasma-sheath interface is shown in Figure 10. Having particle density, ion velocity distribution and electric field calculated from the bulk plasma particle model, the potential drop across the sheath can be calculated for given dielectric material properties (specifically, the secondary electron emission coefficient). Including the sheath acceleration of the ions will increase the ability to predict wall erosion rates.

The dielectric wall effect is taken into account by introducing an effective coefficient of secondary electron emission (SEE). The SEE coefficient is assumed to be a linear function of the electron temperature. In a Hall thruster, there are fluxes of primary electrons to the wall, SEE electrons, and SEE electron fluxes from the opposite wall. Due to partial electron thermalization, effective SEE flux from the opposite wall decreases. This is consistent with earlier predictions. In the steady state total current to the wall must be zero. From the current balance at the dielectric wall one can find that:

$$\Gamma_i = \Gamma_e \left(1 - s + [1 - \alpha]\, s\right) \tag{51}$$

where Γ_i is the ion flux to the wall, Γ_e is the electron flux to the wall, s is SEE coefficient, α is the electron thermalization coefficient that reduces effective SEE (from the opposite wall). Thermalization coefficient is defined as a ratio of the flux of thermalized electrons to the flux of SEE electrons emitted by the wall. For the purpose of this analysis the electron thermalization coefficient is taken as a parameter.

$$\Gamma_i = \Gamma_e \left(1 - s + \beta s\right) \tag{52}$$

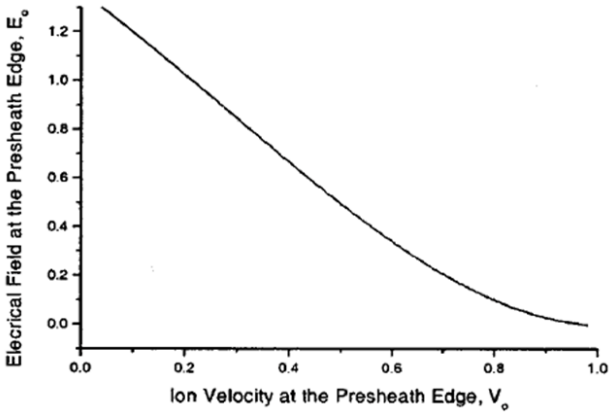


Figure 10. Relationship between boundary conditions at the plasma-sheath interface. Electric field is normalized by Te/LD (LD is the Debye length) and velocity is normalized by the Bohm speed.

where β is the fraction non-thermalized particles. In the case ion flow is calculated based on the Bohm velocity there is a simple analytical expression for the potential drop. The potential drop across the sheath is therefore:

$$\Delta\phi = -T_e \ln\left(\frac{m_i}{m_e}\left[1 - s + \beta s\right]\right) \tag{53}$$

If ion flux is calculated from the PIC simulation the following expression can be used:

$$\Delta\phi = -T_e \ln\left(\frac{n_{e,0}\sqrt{T_e/(2\pi m_e)}}{I_{ion}}\left[1 - s + \beta s\right]\right) \tag{54}$$

s is the SEE total coefficient; β is the fraction non-thermalized particles, which is calculated from the kinetic simulation. The fraction of the electrons that can reach the opposite wall.

Based on the steady state sheath condition we can calculate the potential drop across the sheath. These results are shown in Figure 11, where electron thermalization coefficient was used as a parameter. One can see that the higher thermalization coefficient leads to reduction of the sheath potential drop and as such leads to increase of the electron losses to the wall. On the other hand low thermalization coefficient helps to restore the strong sheath as in the case without SEE.

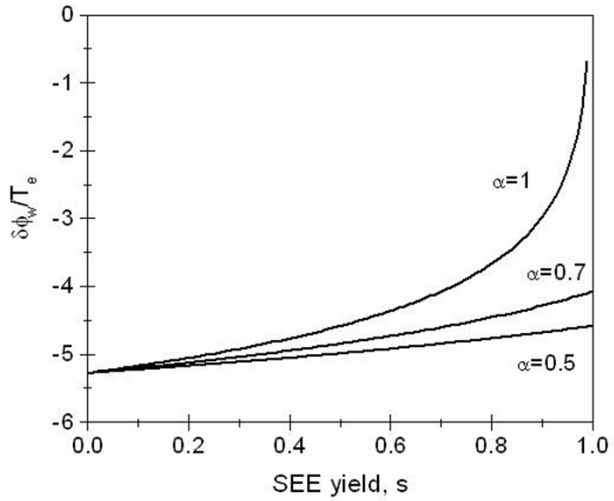


Figure 11. Dependence of the sheath potential drop on the

Additional effect of electron thermalization is on the global discharge characteristics such as bulk electron temperature. To illustrate this effect a 2D plasma flow domain is considered that has lateral boundaries near the dielectric wall. Plasma quasi-neutrality is assumed and therefore the plasma presheath-sheath interface is considered to be the lateral boundary for the plasma flow region. For this demonstration we have employed previously developed hydrodynamic model described in detail elsewhere. The electron energy equation and electron transport are considered in a one-dimensional framework along the centerline. The electron temperature is calculated along the centerline as a balance between Joule heating, ionization and wall losses.

Partial electron thermalization is taken into account parametrically and it was found that it has very strong effect on global discharge characteristic. As an example it is shown in Figure 12 show this effect changes peak electron temperature. It can be seen that higher electron thermalization leads to electron temparture saturation at higher discharge voltage while low electron thermalization produces near linear increase of the electron temperature with discharge voltage. This is very important effect, in particular, when high power and high voltage Hall thrusters are considered.

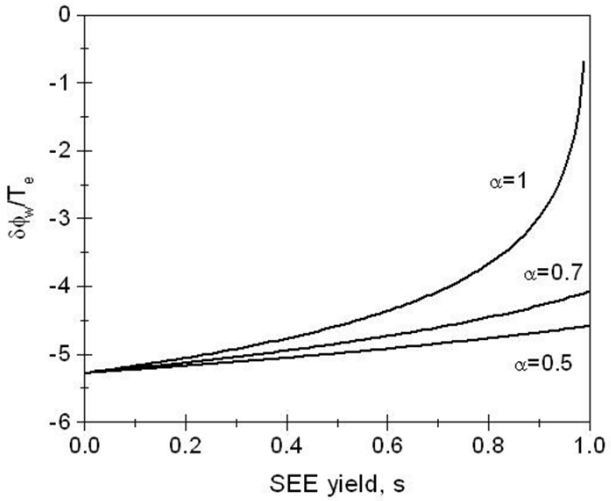


Figure 12. Effect of the partial SEE electron thermalization on electron temperature. $\alpha = 1$ **corresponds to complete thermalization.**

V. Summary of Results

This section presents a summary of important results obtained with components of our multiscale model. We first present results from a Lynx simulation used to investigate electron transport. The important finding from this analysis was that the general linear relationship $\mu_{walls+collisions} = \mu_{walls} + \mu_{collisions}$ does not hold due to synergistic effects. We next shift focus to the 2D sheath code and use it to study the ion dynamics in the presence of an inclined magnetic field. Finally, we present preliminary results from a Starfish simulation of the Princeton Cylindrical Hall Thruster.

A. Synergistic Wall Effects

A common assumption in Hall thruster codes is that

$$\mu = \mu_{classical} + \mu_{walls} + \mu_{Bohm} \tag{55}$$

The classical term corresponds to the transport due to particle collisions. Ignoring the Bohm term, the validity of the $\mu = \mu_{classical} + \mu_{walls}$ relationship can be tested by running three sets of simulations, one with collisions and specular walls, one with wall effects but no collisions, and one with both. The field solver is included to take into account any possible sheath effects. To investigate dependence on temperature, three sets of electron temperatures were also considered, $kT_e = 1$, 10, and 35 eV. A simplified academic approach was used to represent collisions. Instead of utilizing a realistic combination of momentum transfer, Coulomb,

and excitation and ionization collisions, only the momentum transfer interaction was included and a constant cross-section $\sigma = 1 \times 10^{-18}$ m^2 was used. Walls were either specular for the collisions-only case, or were represented by the Sydorenko model. The plasma density in all cases was 1×10^{16} m^{-3} and neutral density was 10^{19} m^{-3}. In all cases $E_\perp = 20,000$ V/m and $B = 0.01$ T were used.

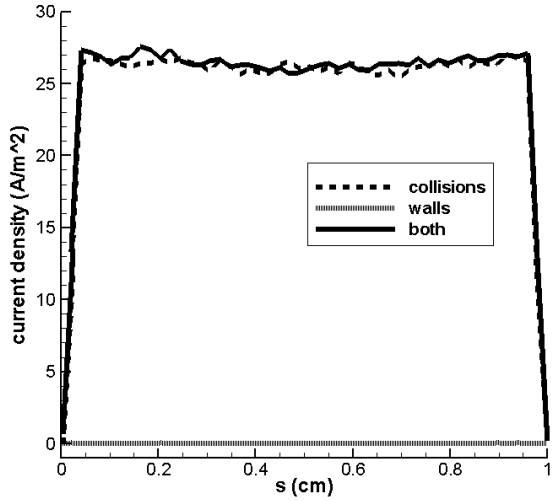
Results from these three runs are shown in Figure 13. The results are grouped by the particle energy. The first column shows the current density variation with the distance along the field line, while the second column shows the corresponding mobility. The second set of pictures also includes a qualitative plot of electron density (in red) and electric potential (in blue). These two curves are included to illustrate the extent of the bulk and the sheath regions. Since the sheath scales with the Debye length, which in turn scales with electron temperature, the sheath thickness is seen to increase from top to bottom. Although values of current density and mobility are included, these are merely illustrative since this particular setup does not correspond to a real thruster. We can see that for $kT_e = 1$ eV, electron transport is completely dominated by collisions. There is very little difference between the results computed with collisions only, and by including both collisions and wall effects. The wall-induced transport is negligible in this case.

The situation changes somewhat as the electron temperature is increased. At $kT_e = 10$ eV, wall interactions can be seen to play a role in mobility, with mobility demonstrating a clear near wall increase. This increase in mobility does not correspond to a linear increase in current density, due to the decrease in electron density in the sheath region. However, of more interest is the observation that $j_{wall+collisions} > j_{wall} + j_{collisions}$. This additional "anomalous" current can be seen in both the bulk region and also in the near wall sheath, where it is especially prominent. This increase in total current is even more pronounced when one considers the $kT_e = 35$ eV case. This case exhibits a finite cross-field current due to wall effects, j_{walls}, with increased magnitude near the walls. On the other hand, the current density due to collisions only shows a flat profile in the bulk region, and rapid decay in the sheath. However, when the two independent processes are combined, the resulting current profile shows a strong deviation from the two bases. A significant increase in near-wall conductivity can be seen when both collisions and wall effects are considered. The bulk population also demonstrates an increase in current. The representative values for $j_{collisions}$, j_{walls}, and j_{both} are 71.5, 1.8 and 83 A/m^2, respectively. Hence, in the bulk population, the combination of the two processes results in a 14% increase in trans-field current density.

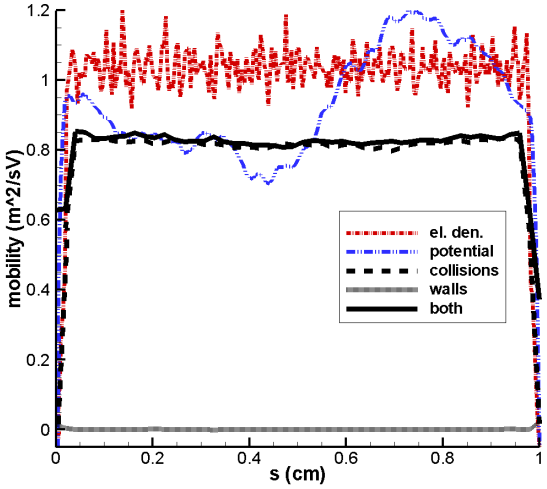
This finding indicates that simple analytical models taking into account each mobility term individually may not account for the interaction between the processes. Such synergistic responses are however not easily included in analytical models, suggesting the need for a kinetic treatment of electron transport. The explanation for this effect is simple. In a steady discharge, a balance develops between the wall potential drop and the parallel component of electron energy. The size of the potential well adjusts self-consistently to trap electrons away from walls. In this classical mode, wall flux is not a significant contributor to conductivity due to the low flux of primary electrons (a different situation arises in the case of a saturated sheath or a magnetic mirror). Collisions scatter the particle velocity components. Even a small angle event, such as Coulomb collision, may impart a sufficient $\Delta v_\parallel$ to allow the electron to reach the wall. This additional scattering is demonstrated in an increased number of secondary electrons. As an example, the case with wall collisions only resulted in a 0.05, 0.80, and 6.81 percent of total population occupied by secondary electrons, where the three values correspond 1, 10, and 35 eV, respectively. Once collisions are included, these percentages increase to 0.06, 1.32, and 10.23%. It should be noted that this additional wall flux is expected to become more pronounced as $kT_{e,\perp} > kT_{e,\parallel}$. This additional energy imparted into the parallel $v_\parallel$ component thus results in an increased number of electrons able to penetrate the sheath and interact with the wall.

B. Effect of Inclined Magnetic Field on Plasma-Wall Interface

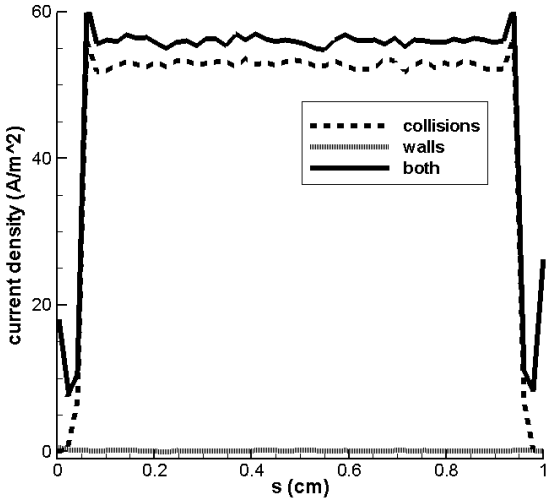
We next discuss results obtained using the two-dimensional sheath code. We used the code to investigate the effect of an inclined magnetic field and the magnetic lens on the motion of ions in the near-wall region. This effort is described in more detail in 33. We ran a simulation for a number of magnetic field incidence angles as well as several magnetic mirror configurations. The simulation started by pre-filling the domain by injecting and propagating ion particles under the initial electric field. This fast pre-fill was found to improve the subsequent solver convergence rate. The simulation then continued in the normal mode in which the electric field was updated at each time step. The simulation continued until steady state characterized by approximately zero net change in particle counts between successive iterations. The simulation then continued for additional 2000 time steps during which results were averaged. The typical number of computational particles at steady state was 700,000. A marching squares algorithm was implemented to automatically con-



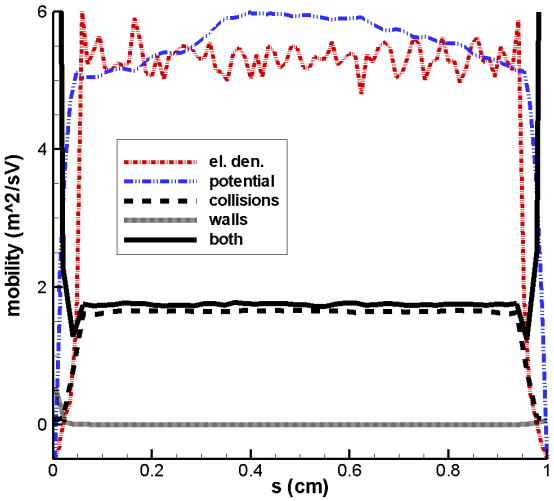
(a) Current density, $kT_e = 1$ eV



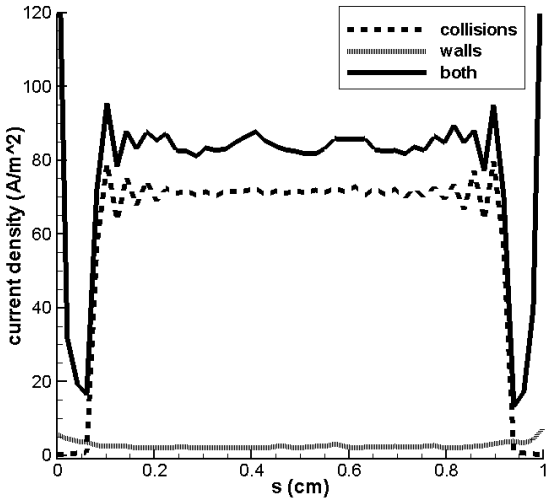
(b) Mobility, $kT_e = 1$ eV



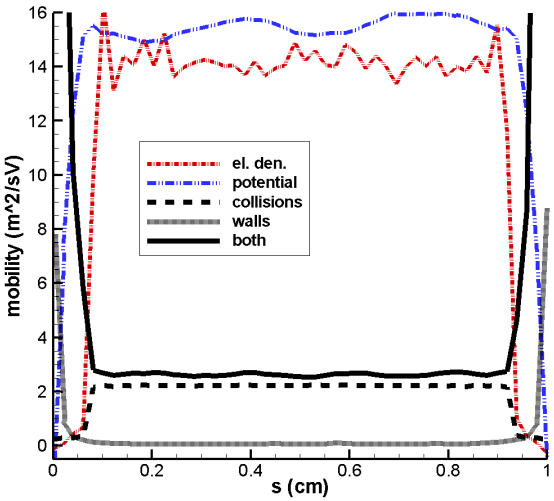
(c) Current density, $kT_e = 10$ eV



(d) Mobility, $kT_e = 10$ eV

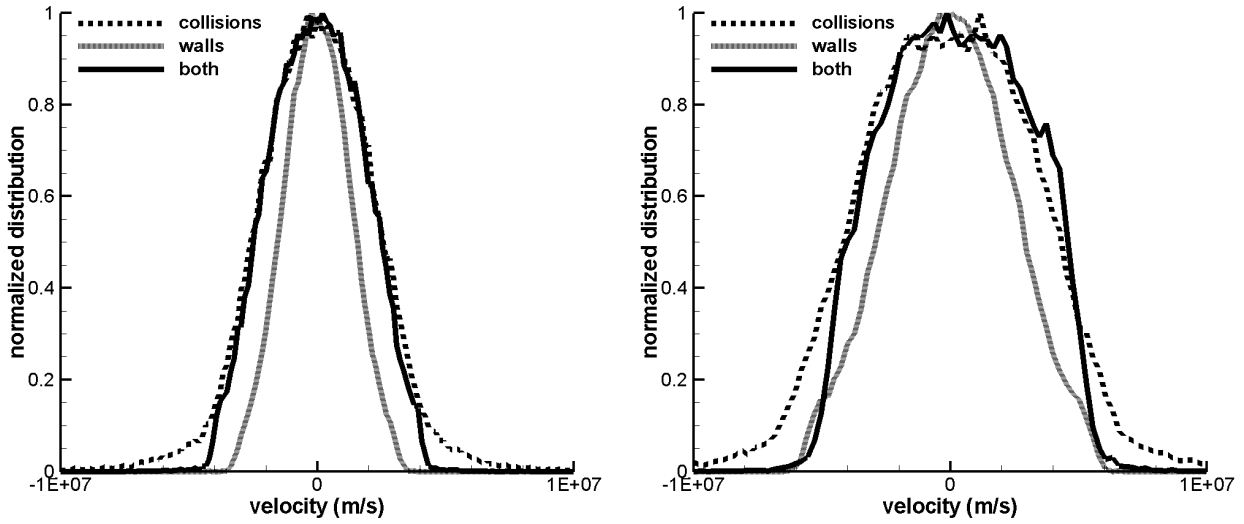


(e) Current density, $kT_e = 35$ eV



(f) Mobility, $kT_e = 35$ eV

Figure 13. Mobilities due to collisions and/or wall effects. Qualitative potentials and electron densities shown for the combined (walls+collisions) case.

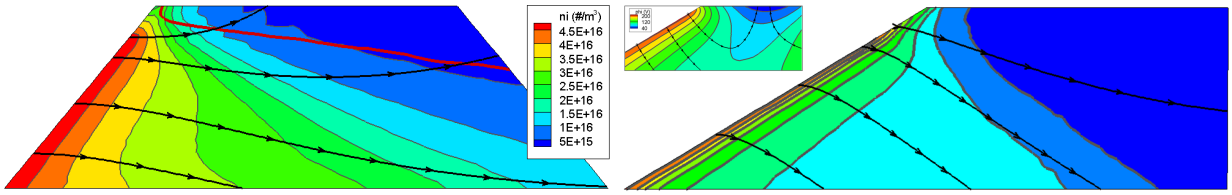


(a) 10 eV (b) 35 eV

Figure 14. Comparison of velocity distribution functions for the three considered cases for initial electron temperatures 10 and 35 eV.

tour the resulting velocity map to obtain the sheath boundary. In this work, we defined the sheath boundary as the contour where the radial component of velocity $v = v_B$, the Bohm velocity.

Figure 15 shows illustrative results obtained for the case of a magnetic mirror with a mirror ratio $R_m = B_w/B_0 = 2$ was used. Cases without and with a 40°magnetic field inclination were compared. The inclined case shown in Figure 15(b) also included a potential "valley" that is known to occur in devices utilizing cusped magnetic profiles. In the CHT, such configurations arise from the difference in the physical location of the inner and outer magnets and the details of the magnetic circuitry. The potential valley was modeled by superimposing a parabolic potential drop over the linear decay, $\phi_0 = \phi_L - E_\perp(z_w - z_{w,0} + \Delta\phi_C\left[4(\hat{z} - 0.5)^2 - 1\right]$, where $\hat{z} = (z_w - z_{w,0})/L_z$ is the normalized distance and $\Delta\phi_C = E_\perp L_z$ is the potential drop in the cusp selected to equal the potential drop in the linear region. The potential profile at steady state is shown in the inset in Figure 15(b). It can be seen that although an electric field forms directing ions into the potential well, this profile has only a negligible effect on the fast moving bulk ions. The primary effect of the potential well is to further increase the electric field accelerating the ions and ions are accelerated towards the channel centerline. It should be noted that this particular example does not take into account the dynamic nature of Hall thruster discharges. Fluctuations in the structure of the internal discharge could lead to the presence of slowly moving ions in the vicinity of the well, and these ions would subsequently be accelerated into the wall.



(a) (b)

Figure 15. Ion density contours in the presence of magnetic mirror. Magnetic mirror strength of 2 is used in both cases. Case (b) includes a 40^o magnetic field inclination and a potential well. The inset shows the potential, with the contours ranging from 200V to 40V.

The ion velocity contours and velocity streamlines at the 85° incidence angle are plotted in Figure 16. The contour plot corresponds to the radial velocity component normalized by the Bohm speed. At this

high incidence angle, the value of the normalized velocity remains below unity, indicating that the Bohm speed is never reached. In addition, ions are moving towards the wall only along a small region near the left boundary. This result is likely a direct byproduct of our loading scheme since it affects only the ions injected into the sheath. Ions originating in the bulk plasma are accelerated away from the wall. Ions located just a small distance from the wall are seen to follow trajectory first parallel to the wall, and subsequently turning away from it. Ions are thus seen to be repelled by the wall, indicating a sheath collapse.

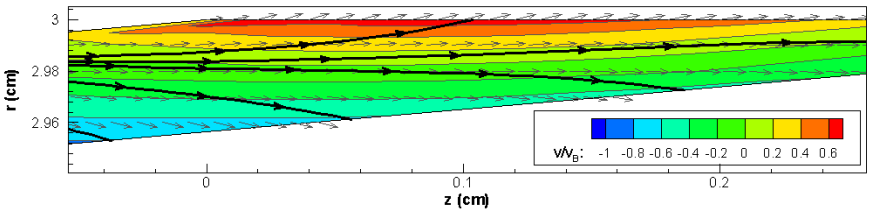
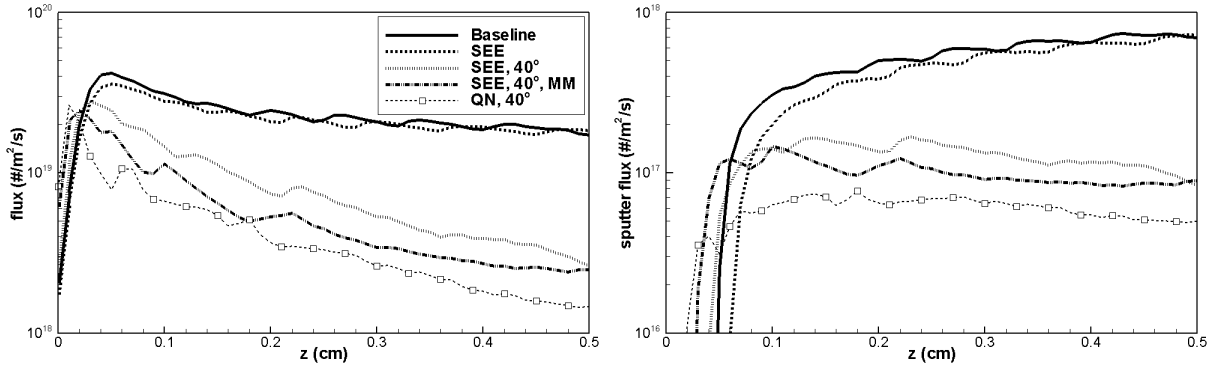


Figure 16. Plots of normalized radial velocity and ion velocity streamlines at $\theta = 85°$.

These numerical results confirm that the presence of highly inclined magnetic fields results in a decreased sheath thickness. This observation is next correlated to the wall flux. From mass conservation, $\Gamma_w = n_{i,s} u_B$, where the terms on the right hand side correspond to the ion density at the sheath edge and the Bohm velocity. The computed wall flux is shown in Figure 17(a). We can see that although the presence of SEE tends to reduce the sheath thickness, it has only a negligible effect on ion wall flux. The flux remains approximately constant for these cases along the wall length, with the slight decrease due to the reduction in bulk ion density due to ion acceleration. The initial spike is an artifact of the loading scheme, as noted previously. The inclined magnetic field is seen to reduce the wall flux considerably, which can be attributed to the net acceleration of ions away from the wall and hence a reduced sheath ion density $n_{i,s} u_B$. This observation has a profound effect on both the ionization efficiency and the thruster lifetime, since ion losses to the walls are a major contributor to both of these inefficiencies. Here we consider only the impact on wall erosion. Material sputtering yield scales with both the impact angle and the energy of the incoming ions. Several models exist for computing sputter yields for Boron Nitride, the material typically used in SPT-type Hall thrusters. In this work we utilize the logarithmic fit suggested by Garnier,[34]

$$Y_0(E) = 0.0156 \ln E - 0.0638 \tag{56}$$



(a) (b)

Figure 17. Comparison of wall flux and emitted sputtered yield for 5 selected configurations.

This fit is valid from the energy threshold of 60 eV up to the keV range. In our analysis we neglect low energy sputtering. For angular dependence of yield, quadratic polynomial fit is recommended by Yim[35]

$$Y(E,\theta) = Y_0 \left[-4.45 \times 10^{-7} \theta^4 + 4.91 \times 10^{-5} \theta^3 - 9.72 \times 10^{-4} \theta^2 + 3.44 \times 10^{-3} + 1 \right] \tag{57}$$

where θ is in degrees and is measured from the wall normal.

Figure 17(b) shows the calculated sputter yield. The profiles for the baseline and SEE configurations are shown to exhibit an increase in sputter yield in the axial direction even though the flux decreases. This response is due to the angular dependence. Similar response is seen in the remaining cases. Presence of a 40° magnetic field inclination results in flux reduction by approximately 60%, leading to a correspondingly similar reduction in erosion rate. In addition, we see that the magnetic mirror has only a marginal effect on wall flux and erosion rates in the inclined field configuration. This can be explained by realizing that both the magnetic field incidence angle and the magnetic mirror generate analogous electric field profiles.

VI. Conclusion

This paper summarizes recent development on a multiscale approach for analyzing Hall thrusters. The approach is based on the thermalized potential model utilized in codes such as HPHall. However, instead of relying on analytical models for the cross-field electron mobility, the mobility is computed self-consistently with a kinetic code Lynx. In addition, a two dimensional code is used to study the ion dynamics in the sheath.

The central component of the multiscale approach is a two dimensional code for the thruster discharge based on the thermalized potential model. In the past, our effort used HPHall as this component. A number of issues prevented continued use of HPHall. As such, we have begun work on a replacement solver Starfish. Starfish is being developed as a general 2D plasma solver with applications not limited to Hall thruster analysis. In addition, we have developed a 2D code for analyzing the dynamics of ions in the sheath in the presence of a magnetic mirror or an inclined magnetic field. The numerical analysis confirms the theoretical prediction that at sufficiently large magnetic field line incidence angles, the ions are repelled from the wall. We have also utilized our kinetic code to study the interaction of electron diffusion processes. Our simulations indicate the presence of synergistic effects. This finding suggests that the common approach of describing mobility by a linear combination of independent terms fails to capture the component of transport due to the interaction between the various processes.

Ackowledgments

Funding for the effort summarized in this paper was provided by The Air Force Office of Scientific Research (AFOSR), the DC Grant Consortium, and the NASA SBIR program. The authors gratefully appreciate the support of these funding agencies.

References

[1]Morozov, A. I., Esinchuk, Y. V., Tilinin, G. N., Trofimov, A., Sharov, Y. A., and Shchepkin, G. Y., "Plasma accelerator with closed electron drift and extended acceleration zone," *Soviet Physics – Technical Physics*, Vol. 17, No. 1, 1972.

[2]Fife, J. and Martinez-Sanchez, M., "Two-Dimensional Hybrid Particle-In-Cell (PIC) Modeling of Hall Thrusters," *24th International Electric Propulsion Conference*, Moscow, Russia, 1995, pp. 1213–1224.

[3]Koo, J. W. and Boyd, I. D., "Modeling of anomalous electron mobility in Hall thrusters," *Physics of Plasmas*, Vol. 13, No. 033501, 2006.

[4]Keidar, M. and Brieda, L., "Modeling electron transport within the framework of hydrodynamic description of Hall thrusters," *44th AIAA Joint Propulsion Conference*, Hartford, CT, 2008, AIAA–2008–5186.

[5]Scharfe, M. K., Thomas, C., Scharfe, D., Gascon, N., Cappelli, M., and Fernandez, E., "Comparison of hybrid Hall thruster model to experimental measurements," *IEEE Transactions on Plasma Science*, Vol. 36, No. 5, 2008.

[6]Hofer, R., Mikellides, I., Katz, I., and Goebel, D., "Wall sheath and electron mobility modeling in hybrid-PIC Hall thruster simulations," *43rd AIAA Joint Propulsion Conference*, Cincinnati, OH, 2007, AIAA–2007–5267.

[7]Sydorenko, D., Smolyakov, A., Kaganovich, I., and Raitses, Y., "Modification of Electron Velocity Distribution in Bounded Plasmas by Secondary Electron Emission," *IEEE Transactions on Plasma Science*, Vol. 34, No. 3, 2006.

[8]Brieda, L., *Multiscale Modeling of Hall Thrusters*, Ph.D. thesis, The George Washington University, 2012.

[9]Raitses, Y. and Fisch, N., "Parametric investigations of a nonconventional Hall thruster," *Physics of Plasmas*, Vol. 8, No. 5, 2001.

[10]Ahedo, E., Santos, R., and Parra, F., "Fulfillment of the kinetic Bohm criterion in a quasineutral particle-in-cell model," *Physics of Plasmas*, Vol. 17, 2010, pp. 073507.

[11]Brieda, L., Keidar, M., Raitses, Y., and Fisch, N., "Self-Consistent Calculation of Electron Transport in a Cylindrical Hall Thruster," *31st International Electric Propulsion Conference*, Ann Arbor, MI, 2009.

[12]Brieda, L. and Keidar, M., "Multiscale Modeling of Hall Thrusters," *31st International Electric Propulsion Conference*, Wiesbaden, Germany, 2011.

[13]Sydorenko, D., *Particle-In-Cell Simulations of Electron Dynamics in Low Pressure Discharges with Magnetic Fields*, Ph.D. thesis, University of Saskatchewan Saskatoon, 2006.

[14]Birdsall, C., "Particle-in-Cell Charged-Particle Simulations, Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC," *Plasma Science, IEEE Transactions on*, Vol. 19, No. 2, 1991, pp. 65–85.

[15]Darnon, F., Garrigues, L., Boeuf, J., Bouchoule, A., and Lyszyk, M., "Spontaneous Oscillations in a Hall Thruster," *IEEE Transactions on Plasma Science*, Vol. 27, No. 1, 1999.

[16]Fox, J., *Advances in Fully-Kinetic PIC Simulations of a Near-Vacuum Hall Thruster and Other Plasma Systems*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[17]Chen, F., *Introduction to Plasma Physics and Controlled Fusion*, Plenum Press, 1983.

[18]Boris, J. and Lee, R., "Non-Physical Self Forces in Electromagnetic Plasma-Simulation Algorithms," *NRL Memorandum Report 2418*, 1972, pp. 1–10.

[19]Szabo, J., *Fully Kinetic Numerical Modeling of a Plasma Thruster*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[20]Lieberman, M. and Lichtenberg, A., *Principles of plasma discharges and materials processing*, Wiley-interscience, 2005.

[21]Bird, G., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Science Publications, 1994.

[22]Particle In Cell Consulting LLC, "Starfish 2D plasma solver," 2012, http://www.particleincell.com/starfish.

[23]Gibbons, M., Kirtley, D., VanGilder, D., and Fife, J., "Flexible Three-Dimensional Modeling of Electric Thrusters in Vacuum Chambers," *39th AIAA Joint Propulsion Conference*, Huntsville, Al, 2003, AIAA-2003-4872.

[24]"Smooth Bzier Spline Through Prescribed Points," http://www.particleincell.com/2012/bezier-splines/.

[25]Fife, J. M., *Hybrid-PIC Modeling and Electrostatic Probe Survey of Hall Thrusters*, Ph.D. thesis, Massachusetts Institute of Technology, 1998.

[26]Lampe, M., Joyce, G., Manheimer, W., and Slinker, S. P., "Quasi-Neutral Particle Simulation of Magnetized Plasma Discharges: General Formalism and Application to ECR Discharges," *IEEE Transcations on Plasma Science*, Vol. 26, No. 6, 1998.

[27]Reid, B. and Gallimore, A., "Plasma Potential Measurements in the Discharge of a 6-kW Hall Thruster," *44th Joint Propulsion Conference*, Hartford, CT, 2008.

[28]Fruchtman, A. and Cohen-Zur, A., "Plasma lens and plume divergence in the Hall thruster," *Applied Physics Letters*, Vol. 89, No. 111501, 2006.

[29]Keidar, M. and Beilis, I. I., "Sheath and boundary conditions for plasma simulations of a Hall thruster discharge with magnetic lenses," *Applied Physics Letters*, Vol. 94, No. 191501, 2009.

[30]Keidar, M., Boyd, I. D., and Beilis, I. I., "Plasma flow and plasma-wall transition in Hall thruster channel," *Physics of Plasmas*, Vol. 8, No. 12, 2001.

[31]Keidar, M. and Boyd, I. D., "On the magnetic mirror effect in Hall thrusters," *Applied Physics Letters*, Vol. 87, No. 121501, 2005.

[32]Dunaevsky, A., Raitses, Y., and Fisch, N., "Secondary electron emission from dielectric materials of a Hall thruster with segmented electrodes," *Physics of Plasmas*, Vol. 10, 2003, pp. 2574.

[33]Brieda, L. and Keidar, M., "Plasma-wall interaction in Hall thrusters with magnetic lens configuration," *Journal of Applied Physics*, Vol. 111, No. 12, 2012, pp. 123302–123302.

[34]Garnier, Y., V., V., Roussell, J. F., and Bernard, J., "Anomalous conductivity and secondary electron emission in Hall effect thrusters," *Journal of Vacuum Science and Technology A*, Vol. 17, 1999, pp. 3246.

[35]Yim, J., Keidar, M., and Boyd, L., "A Hydrodynamic-Based Erosion Model for Hall Thrusters," *29th International Electric Propulsion Conference*, Princeton, NJ, 2005, IEPC–2005–013.